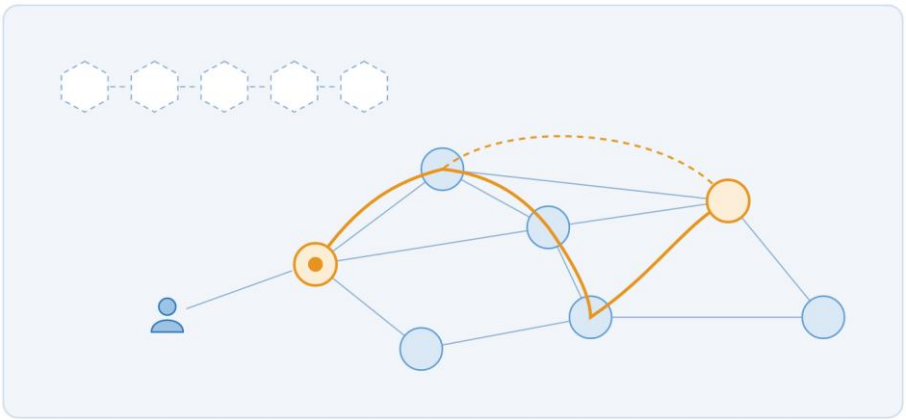
Scrum Manager
Uncovering better ways

Guide

Design Thinking

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 tackling ill-defined problems through design thinking: understanding and reframing the problem, generating alternatives deliberately, making them tangible in order to learn, co-creating with the people affected, and widening the view from the user to the system, 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 Design Thinking 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 Design Thinking evolves slowly at its core, gradually in its reorientation, and very fast in its artificial intelligence layer: the frameworks are fifteen to thirty years old and the techniques of observation, synthesis, ideation, and prototyping rest on decades of practice and evidence; since 2016 its own institutions have reread it as a set of abilities rather than a five-step process, and since 2023 AI has made synthesis, ideation, and prototyping cheap and shifted the value of design work toward deciding which problem matters, who takes part, and what is learned. In this map, CONSOLIDATING marks above all those reorientations that the reference institutions have already made and that mainstream training has yet to absorb. 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 · WHAT DESIGN THINKING IS AND WHAT IT IS FOR

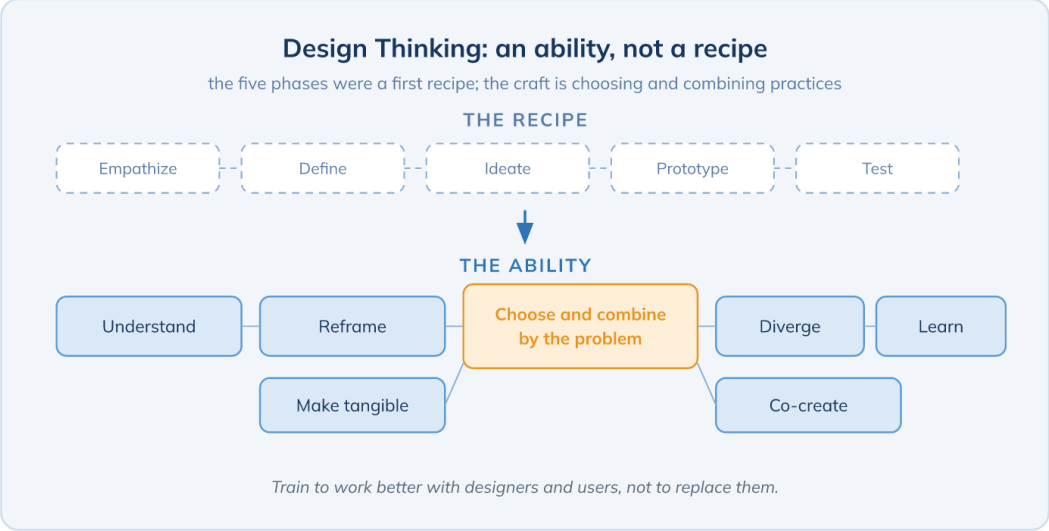
Chapter 1. What Design Thinking is, what it is not, and where it fits ·

ESTABLISHED

An operational definition, its reason for being in ill-defined problems, its porous boundaries, and its place in product work and in agility.

Introduction

Almost everyone who works in product has heard of Design Thinking, and almost nobody defines it the same way. For some it is a five-step process with sticky notes; for others, a consulting fad; for designers, a simplified version of their craft. In this chapter we fix an operational definition, explain why it exists (the problems that do not arrive well defined), and place it in relation to what a Product Owner already knows: user experience, product discovery, and iterative and incremental work. What is learned here is to recognize when a problem calls for this approach and when it calls for something else.



1.1 An operational definition

The definition that professional practice still accepts is the one Tim Brown published in 2008: a discipline that uses the designer's sensibility and methods to match people's needs with what is technologically feasible and with what a viable business strategy can convert into value. From there comes the triad that IDEO, Nielsen Norman Group, and the business schools repeat: a solution has to be **desirable** to people, **viable** for the organization, and **feasible** with the available technology, and design work consists of finding the space where the three intersect.

It helps to know that academia distinguishes two traditions. That of thinking "in the designer's way" comes from Herbert Simon, Donald Schön, Richard Buchanan, and Nigel Cross, and studies how designers reason; the managerial one, that of IDEO, Roger Martin, and the business schools, packaged it for executives and teams who are not designers. The reference academic review reproaches the second for being shallow and not accumulating knowledge. The skill takes the managerial definition as its starting point because it is the one the market recognizes, and corrects it with what the first tradition knows about problems, frames, and expertise.

1.2 Ill-defined problems

Design Thinking exists because of a class of problems. Horst Rittel and Melvin Webber called them "wicked problems" in 1973 and attributed ten properties to them, two of which matter here: they have no definitive formulation, so that formulating the problem is already part of solving it, and every wicked problem can be considered a symptom of another, so choosing at which level to attack it is a decision and not a discovery. Buchanan carried the idea into design in 1992, and Schön had shown earlier that professionals do not receive problems but puzzling situations from which they have to construct the problem, in what he called *problem setting*.

This explains why the approach insists so much on understanding before solving, on returning to the definition, and on testing early. A well-defined problem, with clear success criteria and a known cause, does not need Design Thinking; it needs good engineering or good management. The approach pays off when the problem is ill defined, when the people affected cannot express what they need, and when the first formulation that reaches the team is, almost certainly, a solution in disguise.

To formulate is to solve. In an ill-defined problem, the formulation the team chooses determines the solution space it is going to explore. That is why the work on the problem is not a formality before creativity, but its most important decision.

1.3 What it is not and its boundaries

Design Thinking is not just a brainstorming technique, nor a phase prior to development that ends when the first sprint starts, nor a linear five-step process, nor a promise of innovation. Far from that, it is a way of working on ill-defined problems that is exercised before, during, and after building. Its boundaries with other disciplines are porous, and it is better to describe them precisely than to deny them. With user experience it shares methods, and NN/g itself teaches it as a foundation of UX practice; the difference lies in scope, because Design Thinking covers services, processes, and policies that are not interfaces, and in depth, because UX brings the craft of interaction and evaluation. With product discovery the relationship is part to whole: Marty Cagan says it bluntly when he writes that Design Thinking is a discovery technique, as are the MVP, customer development, or jobs to be done. With service design the boundary is one of object, not of method, because service design widens the view to ecosystems of actors and to organizational implementation.

A practical consequence for this skill: we train people who are not designers to work better with designers and with users, not to replace them. Natasha Jen was right to mock the promise that anyone can be a designer after a two-day workshop, and Jon Kolko dismantled it with the correct argument, that popular Design Thinking never promised to replace the craft, but to make it visible and create demand for depth.

1.4 Where it fits in product work and in agility

Jonny Schneider, from Thoughtworks, gave in 2017 the most quoted formula for the relationship between the three worlds: Design Thinking is how we explore and solve problems, Lean is the framework with which we test our beliefs and learn, and Agile is how we adapt to changing conditions by building software. The temptation is to read it as a sequence (first think, then validate,

then build) and turn it into a waterfall by another name. The correct reading is that of a cross-cutting ability: within a discovery that is continuous, the team turns to design practices when the problem is ill defined, and does so as many times as necessary, also with the product already in production.

The dual-track development that Jeff Patton popularized gives the operational image: a discovery track that learns and validates fast, with prototypes and tests, and a delivery track that builds with production quality, with the same people on both. Design Thinking lives on the first track and feeds the second. And here is the answer to the most repeated criticism of the approach, that of ideas nobody implements: what is learned in a design activity has to enter the opportunity tree, the Product Backlog, and the product decisions, or it will have been for nothing. The Product Owner is the one who decides which challenge is explored and which learning goes to the backlog, and that is why this skill complements their training.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain Design Thinking with Brown's definition and the desirable, viable, and feasible triad, and distinguish the academic tradition from the managerial one.
- Recognize an ill-defined problem by its properties and argue why formulating it is part of solving it.
- Delimit Design Thinking against user experience, product discovery, and service design without denying their overlaps.
- Place it as a cross-cutting ability within continuous discovery and the dual track, and explain how its learning enters the Product Backlog.

Common mistakes and antipatterns

Treating it as a prior phase. A design workshop at the start and nothing afterwards turns the approach into a kick-off ceremony; the problem is reframed also with the product in production.

Applying it to well-defined problems. If the cause is known and the success criteria are clear, engineering or management is what is needed; Design Thinking adds cost without adding learning.

Selling it as innovation. Promising innovation creates expectations that the evidence does not support; what the approach offers is a more reliable way of working on ill-defined problems.

Believing it replaces the designer. The two-day workshop trains people to collaborate with designers and users, not to replace them.

Further reading

· Scrum Manager BoK — Design thinking

· Brown — Design Thinking (HBR, 2008)

· Rittel and Webber — Dilemmas in a General Theory of Planning

· Cagan — The origin of product discovery

· Schneider — Understanding Design Thinking, Lean, and Agile

· Patton — Dual track development is not duel track

BLOCK A · WHAT DESIGN THINKING IS AND WHAT IT IS FOR

Chapter 2. The critique of Design Thinking and what it teaches ·

CONSOLIDATING

Fifteen years of documented critique, the responses of its institutions, the real evidence on its effectiveness, and the whole-approach antipatterns worth recognizing.

Introduction

Few professional practices have received a critique as public and as sustained as Design Thinking, and few have responded to it with such concrete changes. Knowing that critique is not an academic exercise: it is the best vaccine against the antipatterns that ruin the approach in organizations. In this chapter we go through what has been said since 2011, what the d.school, IDEO, and the Design Council have changed in response, what the evidence says about its effectiveness, and what whole-approach antipatterns follow from all of it.

2.1 What has been criticized

The popular critique started from within. Bruce Nussbaum, one of the approach's biggest promoters, wrote in 2011 that companies had absorbed it too well and turned it into a linear, by-the-book methodology. Natasha Jen, of Pentagram, called it in 2017 a buzzword obsessed with a single instrument, sticky notes, and lacking what any design school demands, peer critique (*crit*) and evidence. Lee Vinsel described it as a package that universities sell and that gives "creative confidence without real abilities." Natasha Iskander argued in 2018 that the approach privileges the designer as interpreter of other people's needs and closes ambiguity too early, thereby preserving the status quo it claims to question.

Rebecca Ackermann documented in 2023 the critique that weighs most: the implementation gap. Her cases, IDEO's redesign of San Francisco's school cafeterias and some award-winning sexual health centers in Lusaka that were never scaled, show workshops that generate ideas and consultants who leave before anything changes. Anne-Laure Fayard and Sarah Fathallah summarized three simplifications in 2024: the approach is formulaic, it decontextualizes problems by treating them as individual rather than systemic failures, and it is short-termist because it rewards the proposal and not the implementation. The academic critique (Lucy Kimbell, Johansson-Sköldberg) adds that the managerial discourse ignores the diversity of real design practices and privileges the designer as the main agent.

2.2 What the institutions have answered

The answers have been deeds, not press releases. Stanford's d.school abandoned in 2016 the five-step hexagon as the representation of "the design process" and replaced it with eight abilities; since 2023 it has been retiring empathy as the central concept in favor of care, integrating ethics from the first course, and asking its students to anticipate second- and third-order consequences. IDEO, through Tim Brown, admits that expertise lies more in the hands of those who use the system than in the designer's, lengthens its engagements, and works on building the client's and the community's capabilities while designing. The Design Council responded with the Systemic Design Framework, which adds to the Double Diamond the roles, principles, and activities that the classic process did not contemplate. Kolko, from the profession, agreed with Jen's and Vinsel's diagnosis and separated

the deep practice of design from its popularized version, to which he grants the merit of having made design visible to management.

What the critique teaches. Less theater and more time in the field; abilities and criteria instead of a sequence; real participation of the people affected; peer critique and explicit criteria when deciding; and closing the loop until something changes. Each block of this skill picks up one of those lessons.

2.3 What the evidence says about its effectiveness

It pays to be honest about the evidence. Jeanne Liedtka studied fifty projects over seven years and described plausible mechanisms by which the approach works: immersion counters anchoring on the status quo, design criteria avoid fixation on the first solution, rough prototypes invite interaction, and pre-experiences let users react to what they would not know how to ask for. It is a study of selected cases, with no control group. The most cited academic review, by Micheli and colleagues in 2019, identifies ten attributes and eight tools of the approach and acknowledges that the question of its outcomes remains unresolved. An NN/g study with 87 practitioners found that most know something about Design Thinking but cannot articulate what it involves, and that multiple definitions create friction within teams.

The reasonable conclusion is that Design Thinking has well-described mechanisms and case evidence, and little controlled evidence of outcomes, which is the same situation most agile practices are in. Presenting it that way to a sponsor is more credible than promising a return on investment that nobody has measured.

2.4 The whole-approach antipatterns

From the critique follow the antipatterns that affect the approach as a whole, and which the following chapters detail one by one in their own domain. Innovation theater, in Steve Blank's formulation, is the workshops, classes, and idea marathons that shape culture but rarely deliver anything usable; applied to design, it is the workshop full of sticky notes from which no decision and no learning emerges. The consulting project that ends before implementing. The team that does not share a definition and argues for months about what Design Thinking is and is not. The designer who closes ambiguity too early and turns their interpretation into the users' voice. And, in the age of artificial intelligence, manufacturing evidence about users with synthetic personas, which is the same antipattern of empathy without data with a new tool.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Summarize the main criticisms of Design Thinking (recipe, shallow empathy, unimplemented ideas, designer's privilege, absence of critique) and who made them.
- Explain the changes with which the d.school, IDEO, and the Design Council have responded, and apply them to one's own work.
- Describe honestly the state of the evidence on the approach's effectiveness to a sponsor.
- Recognize the whole-approach antipatterns, starting with innovation theater and the project that ends before implementing.

Common mistakes and antipatterns

Ignoring the critique. Teaching or applying the Design Thinking of 2015, with its five phases and its two-hour empathy, reproduces exactly the failures its own institutions have corrected.

Accepting it whole. The objection that "anyone can be a designer" attacks a promise the approach never made; it pays to tell well-founded critique from caricature.

Promising measured results. The return figures in circulation are marketing; the evidence is of cases and mechanisms.

Debating the definition instead of working. A team that does not share what it means by Design Thinking loses months in friction; agree on the operational definition of chapter 1 and move on.

Further reading

· Ackermann — Design thinking was supposed to fix the world (MIT Technology Review)

· Fayard and Fathallah — Design Thinking Misses the Mark (SSIR)

· Jen — Design Thinking is Bullshit (via Core77)

· Kolko — The Divisiveness of Design Thinking

· Blank — Innovation theater

· NN/g — What Is Design Thinking, Really? (What practitioners say)

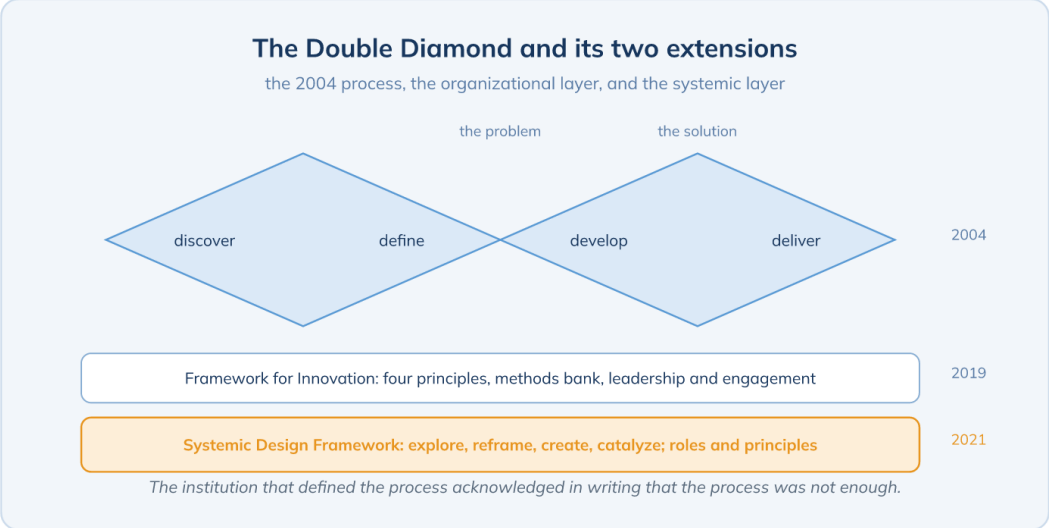
BLOCK B · FRAMEWORKS AS MAPS

Chapter 3. The d.school and the Double Diamond: from phases to abilities and to the system · ESTABLISHED

The two most taught process frameworks, how their own institutions have reread them, and what remains of them as a map.

Introduction

If someone has seen a single image about Design Thinking, it was almost certainly one of these two: five hexagons in a row (empathize, define, ideate, prototype, test) or two joined diamonds (discover, define, develop, deliver). Both are still alive, both are still useful, and both have been relativized by the institutions that created them. In this chapter we present them with that rereading included, because teaching the hexagons without the warning the d.school itself has been giving since 2016 would be teaching a Design Thinking that no longer exists.



3.1 The d.school's five modes and their rereading

Stanford's d.school, founded in 2004 and 2005 with the support of Hasso Plattner, popularized the five modes that almost all training repeats. Empathize is learning from the people you design for; define, formulating a point of view from what was learned; ideate, generating many alternatives; prototype, making some of them tangible; test, putting them in people's hands and learning. Its method deck, the Design Thinking Bootleg of 2018, is still organized by those modes and has a Spanish version under an open license.

In October 2016 Carissa Carter, the school's academic director, published a text with a title that says it all: "Let's stop talking about THE design process." The hexagons, she wrote, "were only a first recipe, a suggestion to get started," and the problem was that they had become the process. As in cooking, the beginner follows recipes and the expert invents them. The d.school then proposed eight design abilities, among them navigating ambiguity, synthesizing information, experimenting rapidly, and designing your design work, which we develop in chapter 25. In 2026 the framework remains active: the school published in March an instrument to measure the ability to navigate ambiguity,

and its institutional presentation is already articulated in four principles (care, ignite, make, adapt), not in five phases.

A first recipe. The d.school still sells a four-day bootcamp organized around the five modes, and at the same time teaches that mastering design consists of no longer needing them. The two things are consistent: the recipe serves to start; the ability, to carry on.

3.2 The Double Diamond and the alternation

The British Design Council was looking in 2003 for a way to describe the design process and its team arrived at four phases: discover, define, develop, and deliver. Drawn as two diamonds, they show what the framework really contributes, which is the alternation between diverging and converging twice: first to understand the problem (opening up to what is there, closing on a definition) and then to solve it (opening up to many solutions, closing on one that is delivered). The model turned twenty in 2023 with millions of references, adoption in organizations such as the British NHS or Meta's labs, and an editorial decision that says a lot about its maturity: the Design Council released it under a Creative Commons license and published a template for collaborative boards.

What gets forgotten when copying it is the institution's own first warning, which is to understand the problem instead of taking it for granted. The first diamond exists because the brief that arrives is almost never the problem that has to be solved; chapter 6 develops that idea.

3.3 The two extensions: organization and system

The Design Council acknowledged in writing, in two steps, that the process was not enough. The Framework for Innovation keeps the four phases and adds four principles (put people first; communicate visually and inclusively; collaborate and co-create; iterate, iterate, iterate), a methods bank organized into explore, shape, and build, and two conditions of the organization without which the process produces nothing: leadership and engagement from those who deliver and receive the ideas. It is the organizational layer, the answer to the implementation gap.

The Systemic Design Framework, presented in 2021 within the Design for Planet mission, is the systemic layer. It keeps the logic of divergence and convergence with four new stages (explore, reframe, create, catalyze), adds four roles (system thinker, connector and convener, designer and maker, leader and storyteller), six principles including zooming in and out and being people and planet centered, and the "invisible" activities that surround the process: orientation and vision setting, connections and relationships, leadership and storytelling, continuing the journey. Block G of this guide rests on it.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the d.school's five modes and the Double Diamond's four phases, and use them as a map of a piece of design work.
- Set out the rereading the d.school itself made in 2016 and why the hexagons are "a first recipe."

- Describe the Double Diamond's alternation of divergence and convergence and the warning to understand the problem before taking it for granted.
- Place the Framework for Innovation and the Systemic Design Framework as the organizational and systemic layers that the process lacked.

Common mistakes and antipatterns

Teaching the hexagons as a sequence. Presenting the five modes as mandatory steps reproduces the mistake the d.school corrected ten years ago.

Skipping the first diamond. Starting to develop solutions on the brief as it arrives is the most common way to solve the wrong problem well.

Ignoring the organizational and systemic layers. A design process without leadership, engagement, or a view of the system produces ideas that nobody implements.

Further reading

· Carter — Let's stop talking about THE design process (d.school)

· d.school — Design Thinking Bootleg

· Design Council — History of the Double Diamond

· Design Council — Framework for Innovation

· Design Council — Systemic Design Framework

BLOCK B · FRAMEWORKS AS MAPS

Chapter 4. IDEO, LUMA, IBM, and Liedtka: systems of methods and questions · ESTABLISHED

Four frameworks that order design work by methods, by skills, by enterprise keys, or by questions, with their current facts.

Introduction

The two frameworks of the previous chapter describe phases. The four in this chapter do something else: they offer catalogs of methods, skills, enterprise keys, or questions, and in doing so they anticipate the skill's thesis, because a catalog is used by choosing. We present them with their 2026 facts, which have changed from what most courses tell, and without developing each method, which appears in the block it belongs to.

4.1 IDEO: three spaces and a field guide

IDEO, the consultancy that popularized the term in the nineties, organizes human-centered design in three overlapping spaces to which one returns again and again: inspiration (learning from people and from the context), ideation (generating, developing, and testing ideas), and implementation (bringing the solution into the world). Its non-profit arm, IDEO.org, published in 2015 *The Field Guide to Human-Centered Design*, with fifty-seven methods, mindsets, and cases, translated into Spanish and free to download, and maintains on Design Kit an online catalog of more than sixty methods filterable by the question they answer. It is the most practical source for a team that is starting out, and its index by question is already a way of choosing according to the problem.

4.2 LUMA: thirty-six methods and recipes

LUMA Institute selected, from more than a thousand, thirty-six human-centered design methods and organized them into three skills with three families each: looking (ethnographic, participatory, and evaluative research), understanding (people and systems; patterns and priorities; problem framing), and making (concept ideation; modeling and prototyping; design rationale). Its pedagogical contribution is the recipe, that is, the combination of a few methods for a specific challenge, which its founder Chris Pacione summed up in a sentence this skill makes its own: design is not a process, it is a discipline that underpins your process. LUMA has been a subsidiary of Mural since 2025, which integrates its methods as board templates, and its home page announces a fourth pillar, aligning, not yet detailed.

4.3 IBM: the loop and the three keys

IBM's Enterprise Design Thinking was born in 2013 to scale design in a company of hundreds of thousands of people, and that is why it adds what the workshop frameworks lack. Its three principles are a focus on user outcomes, restless reinvention (everything is a prototype), and diverse empowered teams. Its process is a non-linear loop of observe, reflect, and make. And its three keys are enterprise mechanisms: Hills, statements of user outcomes (who, what, wow) that align the team without prescribing the solution; Playbacks, periodic meetings with users and stakeholders to review progress; and Sponsor Users, real users who take part in the team throughout the work. A currency warning: in 2026 IBM has withdrawn five of its six badges (only the Practitioner is available,

free of charge) while it rebuilds the program, so it is better to lean on the framework and not on its certification.

4.4 Liedtka: four questions for managers

Jeanne Liedtka, of the Darden School, wrote with Tim Ogilvie the most widely circulated Design Thinking manual for managers, *Designing for Growth*, organized around four questions: what is (exploring the current reality), what if (imagining futures), what wows (choosing what deserves to be tested), and what works (testing in the market). Her Coursera course exceeds four hundred and eighty thousand enrollments, which makes it the most widespread framework among non-designers, and her 2018 article in *Harvard Business Review* presents the approach as a social technology that counters biases, the best defense of its usefulness ever written.

Four ways of ordering the same thing. By spaces, by skills, by enterprise keys, or by questions, the four frameworks speak of the same territory: understanding, imagining, making tangible, and learning. None prescribes a single sequence, and all four are used by choosing.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Describe IDEO's three spaces, LUMA's three skills and recipes, IBM's loop and three keys, and Liedtka's four questions.
- Locate in Design Kit or in the LUMA catalog the method that answers a specific need of the team.
- Apply Hills, Playbacks, and Sponsor Users as mechanisms for scaling design in an organization.
- Cite the current facts of each framework (Field Guide in Spanish, LUMA within Mural, IBM badges under reconstruction) without referring to non-existent certifications.

Common mistakes and antipatterns

Treating the catalogs as task lists. Thirty-six methods are not all applied; two or three are chosen for the challenge, which is what LUMA calls a recipe.

Referring to IBM's badges. Five of the six are withdrawn in 2026; the framework remains current, the certification does not.

Confusing Hills with requirements. A Hill states a user outcome, not a feature; if it prescribes the solution, it stops aligning.

Further reading

· IDEO — Design Kit, methods

· IDEO.org — The Field Guide to Human-Centered Design

· LUMA Institute — LUMA System

· IBM — Enterprise Design Thinking

· IBM — status of the training program

· Liedtka and Ogilvie — Designing for Growth

· Liedtka — Design Thinking for Innovation (Coursera)

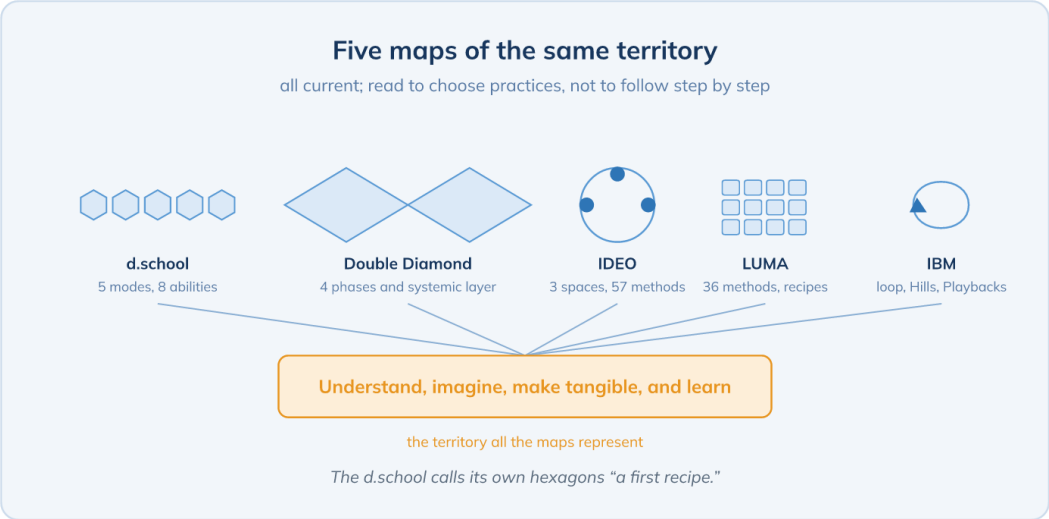
BLOCK B · FRAMEWORKS AS MAPS

Chapter 5. Comparing the frameworks, choosing practices, and the Design Sprint as a format · ESTABLISHED

What the frameworks share and how they differ, when each representation is useful, what the Design Sprint is, and how to move from the maps to the choice of practices.

Introduction

With five frameworks on the table, the useful question is not which one is right, but what they share, how they differ, and what each representation is for. This chapter makes that comparison, presents the Design Sprint as the packaged format organizations ask for most, and closes with the step that gives the block its meaning: reading the frameworks as maps for choosing practices according to what the team needs.



5.1 What they share and how they differ

All the frameworks share four things. They alternate divergence and convergence, opening up to what is there and what could be before closing. They learn from real people in their context, and not from what the team assumes. They make things tangible early, with rough prototypes, to learn before investing. And they iterate, because no idea is finished while something can still be learned from it. They differ along three axes. Some describe phases (d.school, Double Diamond) and others abilities or skills (the d.school's abilities, LUMA). Some offer a sequence (Design Sprint) and others a catalog (IDEO, LUMA). And some were born for the workshop and the project (d.school, IDEO), while others add what is needed in the enterprise (IBM) or in the system (Design Council).

From this follows when each representation is useful. Phases help to explain the work to someone who does not know it and to structure a first project; catalogs, to choose a specific practice for a specific need; IBM's keys, to sustain design in a large organization; the systemic framework, to keep actors and incentives in view. And none of those representations is the design work, just as a map is not the territory.

5.2 The Design Sprint as a format

The Design Sprint that Jake Knapp developed at Google Ventures and published in 2016 packs a complete sequence into five days: on Monday the problem is mapped and a target chosen, on Tuesday everyone sketches alone, on Wednesday the team decides and draws the storyboard, on Thursday a realistic-looking prototype is built, and on Friday it is tested with five customers. It requires a Decider with power, a team of seven people at most, and the discipline of faking the product in a day. AJ&Smart's four-day variant, the only one Knapp has endorsed, compresses the first two days. The Scrum Manager BoK records it with those facts and with its most frequent error, doing it without a Decider who can hold the decision.

The Design Sprint works when the question is how customers react to a specific proposal and the team is already clear about the problem; that is why on Thursday the prototype has an apparent high fidelity. It does not work when the problem is not yet framed, nor for challenges that are too broad, and it is not Design Thinking, but one of its formats. Its relationship with the inception of an initiative, where it may come before or after, is covered by the Agile Inception skill.

5.3 From maps to the choice of practices

The step that closes the block is the one the skill defends from beginning to end: read as maps, the frameworks serve for choosing. Faced with a challenge, the team asks what it needs to understand, imagine, decide, make tangible, or learn, and chooses the practice that answers that need, whether it comes from IDEO's catalog, LUMA's recipes, or a card from the Bootleg. Chapter 24 develops that criterion of choice; blocks C to F teach the practices to choose from.

The territory. Understanding, imagining, making tangible, and learning is what all the maps represent. When a team masters those four things, the framework it uses is a matter of shared vocabulary.

What you should be able to do

Having mastered this chapter, a professional is able to:

- State what the Design Thinking frameworks share and the three axes along which they differ.
- Choose which representation is appropriate for explaining, structuring, scaling, or widening a piece of design work.
- Describe the Design Sprint, its conditions, and its limits, and decide when it is the right format.
- Move from frameworks as maps to the choice of practices according to what the team needs to understand, imagine, make tangible, or learn.

Common mistakes and antipatterns

Looking for the right framework. Arguing about which one is right is wasted time; they all describe the same territory and are chosen for what they help to explain or to do.

Running a Design Sprint without a framed problem. The format presupposes a clear problem and a question about customers' reaction; without that, the week produces a prototype of the first idea.

Confusing the format with the approach. The Design Sprint is a packaged week; Design Thinking is the ability that allows deciding whether that week is what is needed.

Further reading

· Scrum Manager BoK — Design sprint

· GV — The Design Sprint

· Knapp — Design Sprint guide

· LUMA — Recipes: the power of combining design methods

· Skill Arena — Agile Inception

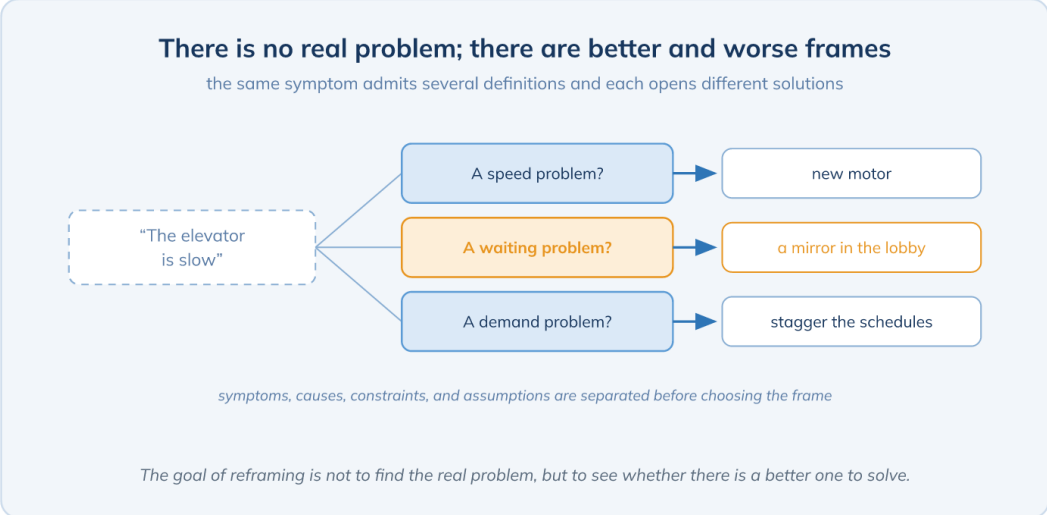
BLOCK C · UNDERSTANDING AND REFRAMING THE PROBLEM

Chapter 6. The stated problem and the better frame · ESTABLISHED

Symptoms, causes, assumptions, and solutions in disguise; why there is no "real problem" and how a problem is reframed before ideating.

Introduction

The problem that reaches a team is almost never the one that has to be solved. It arrives as a symptom ("users abandon the sign-up"), as a solution in disguise ("we need an app"), or as an inherited formulation nobody has reviewed. The first job of Design Thinking is on that formulation, and in this chapter we will see why, with the evidence that supports it, and with which concrete practices a problem is reframed before starting to ideate.



6.1 Symptoms, causes, assumptions, and solutions in disguise

Behind a stated problem it helps to separate four things. The **symptoms**, which are what is seen and what usually motivates the brief. The possible **causes**, which are almost always several. The **constraints**, which limit what can be done and are sometimes what actually produces the symptom. And the **implicit assumptions**, which are the unspoken beliefs about who has the problem and why. Rittel and Webber observed that every wicked problem can be considered a symptom of another, and from there come a temptation and a warning: going up a level ("the problem is not the sign-up, it is trust in the brand") opens up more solutions, but the higher one goes the more general and the more intractable the problem becomes. Choosing the level is a decision with a cost.

The solution disguised as a problem has a simple test that IDEO proposes when framing a challenge: if the team can propose five different solutions in a few minutes, the statement is well posed; if it admits only one, it was a solution. Teresa Torres adds another clue from product discovery: when the "opportunity" someone brings already names a feature, it is not an opportunity.

6.2 There is no "real problem"

The idea that beneath the symptom lies a real problem that merely needs digging up is intuitive and misleading. Thomas Wedell-Wedellsborg formulated it in *Harvard Business Review* in 2017 from a

survey of 106 executives in 17 countries, 85 % of whom acknowledged that their organization was bad at diagnosing problems: the very idea of a single root problem can be misleading, because problems are multi-causal and admit many approaches, and the goal of reframing "is not to find the real problem, but to see whether there is a better one to solve." His canonical example comes from Russell Ackoff: tenants complain that the elevator is slow, and instead of a new motor someone proposes putting mirrors in the lobby; the mirror does not make the elevator faster, it proposes a different understanding of the problem, which was the boring wait and not the speed.

There is experimental evidence of why deliberate reframing is needed and wanting to see alternatives is not enough. Merim Bilalić and colleagues showed with chess players that, once a solution was found, experts said they were looking for a better one while their eyes stayed fixed on the features of the board related to the first; this is the Einstellung effect, which reduces an expert's capacity to that of a player three levels below. Attention does not change if the representation of the problem does not change.

A better frame. Reframing is not diagnosing the root cause. It is changing the definition of the problem so that solutions appear that the previous definition did not allow one to see. The useful question is "is there a better problem to solve?", not "what is the real problem?".

6.3 The reframing practices

Wedell-Wedellsborg proposes seven practices that work in a meeting of less than an hour. Legitimizing the question, that is, agreeing with the group that time will be spent discussing the problem before solving it. Bringing in outsiders, people from other areas who do not share the team's assumptions, and asking them for input, not solutions. Asking for written definitions before the meeting, in full and anonymous sentences; almost none coincide, and almost none include the word "I." Asking what is missing from the story. Considering several categories (is it a problem of incentives, of expectations, of attitude, of usability?). Analyzing the positive exceptions, the cases where the problem does not appear. And questioning the objective, because sometimes the open window and the draft are solved by the door next to it.

The five whys deserve a caveat. They serve to climb the ladder of causes of a problem already defined and are useful for that, but they presuppose a linear chain and a single cause, and they stop wherever the team wants; they dig deeper into the frame that is already there instead of changing it. The abstraction ladder of chapter 9, which climbs with "why" and descends with "how," does the same in both directions and without presupposing a root.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Separate, in a stated problem, the symptoms, the possible causes, the constraints, and the implicit assumptions.
- Detect a solution disguised as a problem with the five-solutions test and reframe it as an open problem.
- Explain why there is no single real problem and what reframing seeks, with the elevator example.

- Apply the reframing practices (written definitions, outsiders, categories, positive exceptions, questioning the objective) and know the limit of the five whys.

Common mistakes and antipatterns

Accepting the brief as it arrives. The stated problem is usually a symptom or a solution; starting to ideate on it is solving the wrong problem well.

Looking for the real problem. Digging down to a single root cause with the five whys deepens the existing frame; reframing looks for a different frame.

Going up too many levels. "The problem is the company culture" opens up every solution and allows none to be started; the level is chosen by what can be done.

Reframing without checking. Wedell-Wedellsborg himself warns: reframing goes hand in hand with observation and real-world testing, it does not replace them.

Further reading

· Wedell-Wedellsborg — Are You Solving the Right Problems? (HBR)

· Bilalić, McLeod, and Gobet — the Einstellung effect

· IDEO — Frame your design challenge

· Torres — Opportunity solution tree

· Rittel and Webber — Dilemmas in a General Theory of Planning

BLOCK C · UNDERSTANDING AND REFRAMING THE PROBLEM

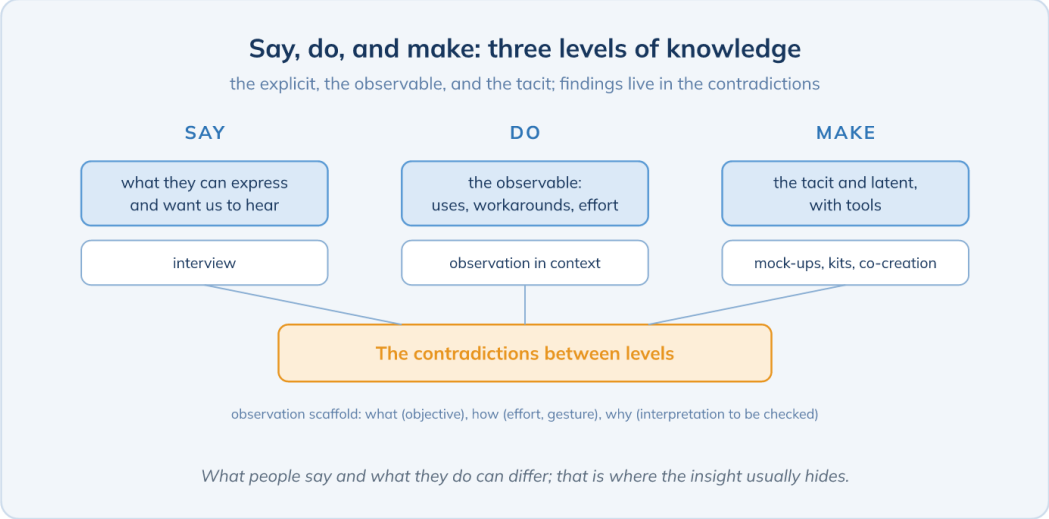
Chapter 7. Observing with intent and empathizing within limits ·

ESTABLISHED

What people say, do, and make; the observation scaffold that belongs to design; and the three documented limits of empathy, with their answer.

Introduction

User research techniques (interviewing, observing in context, keeping diaries, following someone through their day) are already part of Product Owner training and of Scrum Manager's Discovery and customer validation skill, and here we take them as known sources of evidence. What Design Thinking adds is something else: a way of looking, a scaffold for separating what is observed from what is interpreted, a criterion for choosing whom to observe, and an idea of empathy that has changed a great deal in recent years. That is what this chapter is about.



7.1 Say, do, and make

Elizabeth Sanders fixed in 2002 a distinction that practice repeats without citing it. Listening to what people **say** gives us their explicit knowledge, and only what they want us to hear. Observing what they **do** and use gives us observable information, including the workarounds with which they solve what the product does not. And seeing what they **make** when given tools to express themselves gives us access to the tacit and to latent needs. Design work is interested above all in the contradictions between the three levels, because that is where the findings usually hide; the d.school's Bootleg says it simply in its card on interviews: "look for inconsistencies; what people say and do can be different."

The most useful observation scaffold for someone who is not a researcher is what the d.school calls **what, how, and why**. First the objective is noted, what the person does, without assuming anything. Then the how, with what effort, with what gesture, with what expression. And only at the end does one jump to the branch of interpretation, the why, which produces assumptions that have to be checked with the people themselves. It is a scaffold that forces the separation of observation and inference, the same discipline the Discovery skill teaches when taking notes. Other field templates,

such as AEIOU (activities, environments, interactions, objects, users) or the shadowing that the Scrum Manager BoK records, serve the same function.

7.2 Whom to observe: extremes and analogues

To inspire, the sample does not seek to represent but to reveal. **Extreme users**, those who use the product a lot, very little, or in a strange way, have amplified needs and visible workarounds that go unnoticed in the middle of the distribution, and their needs usually coincide with those of a wider population; IDEO recommends talking to the extremes and to those in the middle. **Analogous contexts** are another resource specific to design: to understand time pressure in an operating room one observes an airport runway, because it shares the attribute of interest without sharing the context. Neither technique replaces the representative sample when what is sought is to measure; they serve to see.

7.3 The limits of empathy

Empathy was for years the central word of the approach, and the academic critique of 2018 to 2023 has put it in its place. Three limits are documented. The first is epistemic: we project our experience onto others (the false-consensus effect, which NN/g sums up for designers as "you are not the user") and we fixate on the first interpretation. The second is ethical: Cynthia Bennett and Daniela Rosner showed in 2019 that empathy techniques can distance the designer from the people they claim to approach; whoever puts on a blindfold to "understand" blindness learns about their own experience with the blindfold, and they propose moving from "being like" to "being with." The third is methodological: an empathy map or a persona filled in without data is a list of the team's assumptions in research format, and NN/g warns that an empty or sparse quadrant means more research is needed, not more imagination.

The convergent answer is not more empathy, but real contact and participation: users who are part of the team, like IBM's Sponsor Users, and affected people who speak for themselves, as the Design Justice Network asks. The d.school itself has removed the empathy map from its 2018 Bootleg and replaces "empathy" with "care" in its teaching. If the empathy map is used, the version Dave Gray published in 2017 is advisable, which starts from the person's goal and leaves what they think and feel for the end, as an inference from what they see, say, do, and hear, and always on data, as the BoK page reminds us.

From being like to being with. The synthetic personas that artificial intelligence generates are the recent version of the same limit: they serve to rehearse questions and formulate hypotheses, never as evidence about real people. Chapter 27 develops why.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Distinguish what people say, do, and make, and look for the contradictions between those levels as a source of findings.
- Apply the what, how, and why scaffold (or an equivalent template) to separate observation and interpretation.

- Choose extreme users and analogous contexts to inspire, and know when a representative sample is needed.
- Explain the three limits of empathy and replace it with real contact and participation when the team starts speaking on behalf of users.

Common mistakes and antipatterns

Filling in the empathy map in the meeting room. Without field data, the map records the team's assumptions with the look of research; the BoK flags it as the technique's most frequent error.

Believing you are the user. False consensus is part of human psychology; observing real people is the only correction.

Simulating someone else's experience. Blindfolds, borrowed wheelchairs, or fictional personas teach about whoever uses them, not about whoever lives that condition.

Jumping to the why. Interpreting before noting what and how produces findings nobody can check.

Further reading

· Sanders and Stappers — Co-creation and the new landscapes of design

· d.school — Design Thinking Bootleg (What? How? Why?; Extreme users)

· IDEO — Extremes and mainstreams

· NN/g — You are not the user: the false-consensus effect

· NN/g — Empathy mapping

· Scrum Manager BoK — Empathy map (Mapa de empatía)

· Design Justice Network — Principles

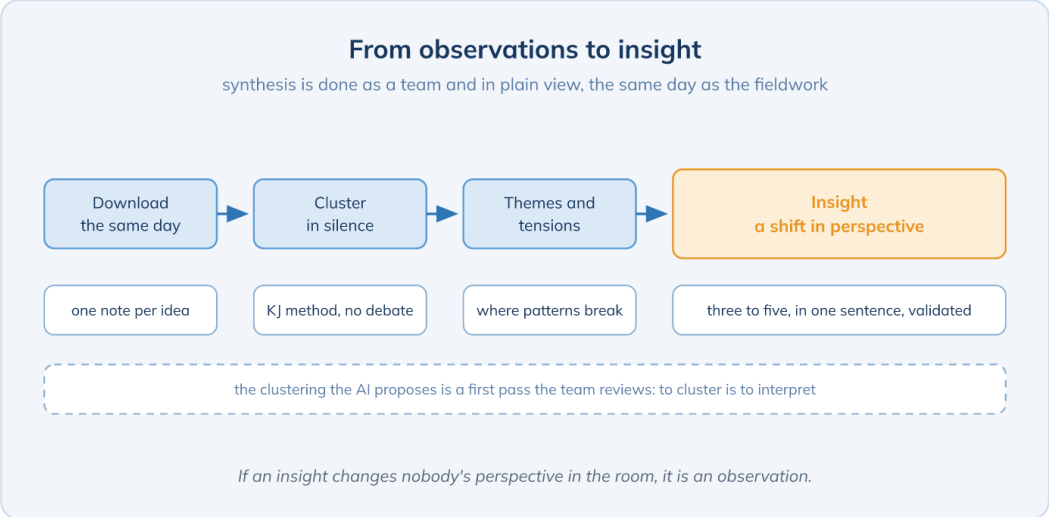
BLOCK C · UNDERSTANDING AND REFRAMING THE PROBLEM

Chapter 8. Collective synthesis: from observations to insight · ESTABLISHED

The invisible step of design done as a team and in plain view: download, affinity, themes, tensions, and insights, and what AI does in it.

Introduction

Between fieldwork and problem definition there is a step almost nobody budgets for because almost nobody sees it: synthesis. Jon Kolko called it the "magic box" of design, because it happens in private and only the result is seen, and that is why clients dismiss research as waste and consultancies set no time aside for it. Design Thinking responds by doing synthesis as a team and in plain view. In this chapter we see how it is done, what an insight is and what it is not, and what role artificial intelligence plays in this step.



8.1 Downloading and sharing stories

Synthesis starts the same day as the fieldwork, while impressions are fresh. IDEO calls it downloading your learnings: in a circle, each team member takes turns to pour out whom they met, what they saw, what data and what impressions they take away, one idea per note, while the others listen and ask. The d.school does the same with story sharing, and observes that by listening and asking, colleagues draw out nuances and meanings that the person who was in the field had not noticed; that is where synthesis starts. Half an hour per person is enough. The Discovery skill teaches the prior discipline, separating fact, quote, and interpretation in the notes; what matters here is the collective ritual.

8.2 Clustering in silence and reading the themes

Affinity clustering comes from the KJ method of Jiro Kawakita, a Japanese ethnographer who in the sixties discovered, with his field notes from Nepal spread out on the table, that the spatial arrangement of the cards allowed him to see new meanings. The version design teams use keeps his most important rule: clustering is done in silence, without debate, letting the notes find their neighbors, and only afterwards is each group named and discussed; the Scrum Manager BoK records

it that way. NN/g adds three warnings: do not force groups, do not over-categorize, and do not discard a small group just because it has few notes, because exceptions point to segments.

From the groups come themes, which NN/g defines as a belief, practice, or need that appears several times across participants or sources, and with the themes it pays to look for the tensions, that is, the points where an expected correlation breaks or where two observations contradict each other, because that is usually where what the team did not know lies. The user journey map is also a synthesis tool, and the d.school recommends being exhaustive with it because what seems insignificant may be the piece that becomes a finding.

8.3 What an insight is and what it is not

IDEO rates insight statements as the only hard method of its ideation phase, and rightly so. An **insight** is not an observation or a piece of data; IDEO U defines it as a shift in perspective that reveals an unmet need or opens a new way of thinking, and it must inform, inspire, and be memorable. The d.school gives the negative test in its review matrix: a shallow insight is what the team already knew before starting, and a deep one exposes the problem in a new light. The most practical technique for reaching it is the leap from surprise to insight: what surprised is noted down and "I wonder if this means that the user…" is completed, in rounds. A team that leaves synthesis with three to five insights formulated in one sentence, validated with someone outside, has what it needs to define the problem; one that leaves with twenty groups of notes has not synthesized yet.

What the team already knew. If an insight does not change anyone's perspective in the room, it is an observation. The test is simple: was anyone surprised?

8.4 With AI

Research tools and collaborative boards cluster notes and propose themes automatically, and the evidence says what they do well: coding with given criteria and offering a first inductive pass that an analyst reviews. What they do not do is interpret latent meanings, tensions, and contradictions, which is exactly where insights are born, and NN/g rates their analysis as shallow, because it captures word patterns and not deep connections. Mural acknowledges that two people would cluster the same wall of notes differently, and the Scrum Manager BoK warns that skipping the discussion when clustering removes the central value of the technique, which is the cognitive alignment of the team. The practical rule follows from there: automatic clustering is a first-pass proposal that the team reviews in plain view, never the synthesis, because to cluster is to interpret. Chapter 26 develops the underlying question, who leads at each moment.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Organize the immediate download and story sharing after fieldwork.
- Facilitate a silent affinity clustering and afterwards read the themes and tensions with the team.
- Formulate three to five insights that change the team's perspective and distinguish them from observations and from what was already known.

- Use AI's automatic clustering as a first-pass proposal without delegating synthesis to it.

Common mistakes and antipatterns

Not budgeting for synthesis. It is the invisible step; if it has no time allotted, the team goes from the field to ideation without having understood anything.

Debating while clustering. The KJ rule is to cluster in silence; premature debate imposes the interpretation of whoever talks most.

Taking the group of notes for an insight. Twenty named groups are not findings; synthesis ends in sentences that change the perspective.

Accepting the AI's clustering. It is one possible reading among many; if the team does not discuss it, it has not synthesized.

Further reading

· Kolko — Abductive thinking and sensemaking

· IDEO — Download your learnings

· IDEO — Create insight statements

· NN/g — Affinity diagramming

· NN/g — Thematic analysis

· Scrum Manager BoK — Affinity Mapping

· Skill Arena — Discovery and customer validation

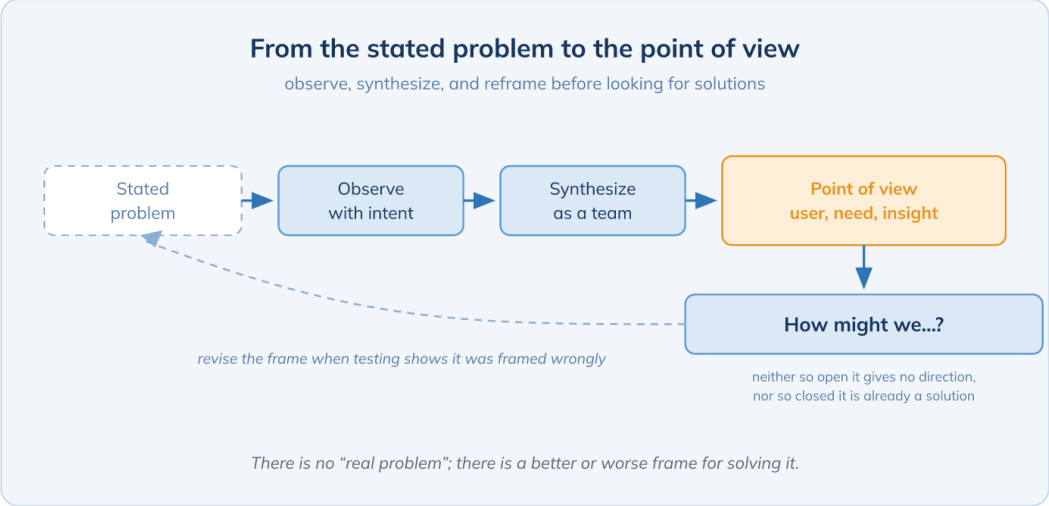
BLOCK C · UNDERSTANDING AND REFRAMING THE PROBLEM

Chapter 9. Defining and reframing: the challenge, the point of view, and "How might we…?" · ESTABLISHED

The challenge set before researching, the point of view formulated afterwards, the generative questions, the abstraction ladders, and the criterion for choosing which challenge to explore.

Introduction

Defining the problem is the moment when the work on the problem becomes an artifact the team can share, discuss, and revise. The three reference programs agree on two artifacts, the point of view and the "How might we…?" question, and on one rule, that the definition is revised throughout the cycle. In this chapter we present them with their rules, add the abstraction ladders for moving between the general and the actionable, and close with the criterion for choosing which challenge to explore, which is a decision of the Product Owner.



9.1 The challenge first, the point of view afterwards

There are two moments of definition, and they should not be confused. The **design challenge** is formulated before researching and delimits the scope: IDEO asks that it point to the ultimate impact, take context and constraints into account, and allow a variety of solutions, and that it be revised. The **point of view** (POV) is formulated after synthesizing, and it is the definition of the problem that the research has produced. In its classic form it brings together user, need, and insight ("the mother traveling alone with three children needs to keep them entertained while waiting because a delayed flight turns the airport into a threat"); the 2018 Bootleg template writes it in four steps: we met, we were surprised to notice, we wonder if this means, it would be game-changing to. A good point of view keeps the emotion and the individual, includes a strong insight, is understood without explanation, and dictates no solution.

9.2 "How might we…?"

The "How might we…?" (HMW) question turns the point of view into a space for exploration. It has a documented origin: Min Basadur, at Procter & Gamble in the early seventies, replaced "how can we?" with "how might we?" so that the team would stop judging feasibility while generating, and

the Coast brand came out of it; Charles Warren took it to IDEO and Tim Brown adopted it. Its rules are about breadth. The d.school teaches them with an example: between the too narrow ("how might we create a cone that doesn't drip?") and the too broad ("how might we redesign dessert?") lies the useful one ("how might we make ice cream more portable?"). From one point of view come several questions along different paths, easing the tension, exploring the opposite, questioning an assumption, creating an analogy, and the five-solutions test says whether the question is well posed. The Scrum Manager BoK fixes the Spanish vocabulary and its frequent error, generating questions without prior research.

Neither open nor closed. A question that is too open gives no direction and one that is too closed is already a solution. The team knows it got it right when the question lets it start ideating and leaves room for ideas nobody expected.

9.3 Abstraction ladders and revising the frame

To move between the general and the actionable, the abstraction ladder serves, which the d.school calls *why-how laddering* and LUMA *abstraction laddering*. Asking "why" climbs: the statements become more meaningful and less actionable, until they reach basic human needs. Asking "how" descends: they become more concrete and closer to a solution. The work consists of finding the intermediate rung, meaningful and actionable at the same time. It is the same mechanism as the five whys, but in both directions and without presupposing a single cause.

The final rule is that the definition does not close. The Bootleg's testing card says it bluntly: testing a prototype can reveal not only that the solution was wrong, but that the problem was framed wrongly. A team that treats its point of view as a revisable hypothesis returns to it after each test; one that treats it as a requirement has turned Design Thinking into waterfall.

9.4 Choosing which challenge to explore

With several points of view and questions on the table, someone has to decide which one is explored, and that decision belongs to the Product Owner. The literature offers no canonical list of criteria; combining what IDEO says (ultimate impact, context, and constraints), the d.school (depth of the insight), and continuous discovery (potential to move the outcome pursued), at Scrum Manager we propose five: relevance to people, uncertainty it reduces, potential impact, learning capacity, and relationship to the product objectives. The last two connect the choice of challenge with the opportunities of discovery and with the Product Goal, which Product Owner training already covers; opportunity prioritization is taken as known.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Formulate a design challenge before researching and a point of view after synthesizing, and distinguish both moments.
- Write "How might we…?" questions with the right breadth and check them with the five-solutions test.
- Climb and descend the abstraction ladder to a rung that is meaningful and actionable.

- Revise the definition of the problem after testing and choose which challenge to explore with explicit criteria.

Common mistakes and antipatterns

Jumping from synthesis to brainstorming. Without a written point of view, each person ideates on their own version of the problem.

Questions that are already solutions. "How might we create an app that…?" is not a generative question; the five-solutions test detects it.

Closing the definition. Treating the point of view as a requirement prevents learning from the test the most valuable thing, that the problem was framed wrongly.

Choosing the challenge by enthusiasm. Without explicit criteria the challenge of whoever insists most gets explored; the five criteria make it debatable.

Further reading

· Scrum Manager BoK — How might we

· d.school — Design Thinking Bootleg (POV, HMW, why-how laddering)

· Basadur — The origin of How Might We

· IxDF — Define the problem and interpret the results

· LUMA — Abstraction laddering

· IDEO — Frame your design challenge

BLOCK D · DIVERGING WITH METHOD AND CONVERGING WITH CRITERIA

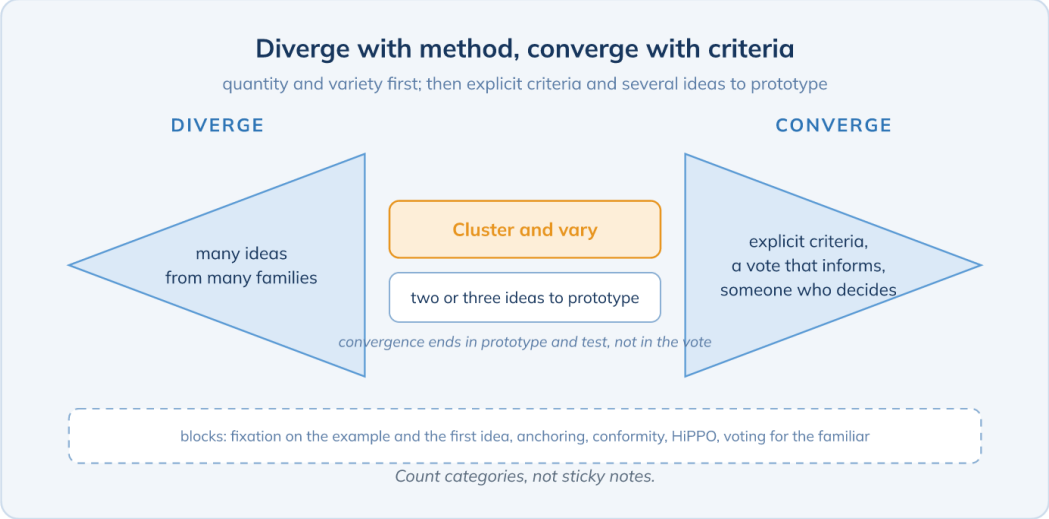
Chapter 10. Diverging and converging: variety, blocks, and fixation ·

ESTABLISHED

Separating generation and evaluation, measuring variety and not just quantity, and knowing the blocks by name and evidence, starting with design fixation.

Introduction

Ideation is the part of Design Thinking everyone thinks they know and the one done worst. This chapter lays the foundations: where the rule of separating generation from evaluation comes from, why the number of ideas is not enough if variety is missing, and which blocks, with names and with evidence, make a group produce twenty variations of the first idea while believing it has diverged. The next chapter teaches the way of working that the evidence supports; this one explains what it works against.



10.1 Diverging and converging, and measuring variety

The distinction between divergent and convergent thinking is J. P. Guilford's, from the fifties, and the rule of separating generation from evaluation belongs to Alex Osborn, the advertising executive who invented brainstorming in 1953, and to the Creative Problem Solving method he developed with Sidney Parnes. Design Thinking inherited both without citing their authors: the Double Diamond alternates divergence and convergence twice, and the d.school's Bootleg sums up that "a common theme across all ideation techniques is deferring judgment."

What training usually forgets is the vocabulary for measuring. Paul Torrance operationalized divergence in four measures: **fluency** (how many ideas), **flexibility** (how many distinct categories), **originality** (how many rare ones), and **elaboration** (how much detail). Quantity is only the first. A session with eighty ideas in four families has diverged less than another with thirty in twelve, and that is why the practical rule of this skill is to count categories, not sticky notes. Girotra, Terwiesch, and Ulrich add from innovation management two more measures, the variance of quality and the group's ability to recognize its best ideas, because what an organization wants is not a hundred good ideas but one outstanding one.

10.2 Design fixation

The best documented block in the discipline is **design fixation**, which David Jansson and Steven Smith described in 1991: when those who are going to design are shown an example, their proposals repeat the example's features, even when they are warned that those features are flaws, and they do so without realizing. It happens in students, in professionals, and, as Julie Linsey showed in 2010, in engineering professors who moreover do not notice it happening to them. Kathryn Leahy and colleagues discovered something more uncomfortable in 2020: we fixate even more on our own first idea than on someone else's example, and we prefer our initial designs to other people's.

Expertise does not immunize. Bilalić called it "reality and myth" when studying chess players: the Einstellung effect affects all levels and is noticed less with experience, although the most expert also have more resources to get out of it. Nathan Crilly, after interviewing professional designers, found the only defense that works: recognizing episodes of fixation when they occur and reflecting on them, something that is learned from the experience of having seen varied solutions and not from that of having failed.

We fixate on our first idea. The 2020 evidence changes an intuition: someone else's example fixates, but one's own first idea fixates more. That is why really diverging requires techniques that force abandoning it, not just the will to do so.

10.3 The other blocks

The rest of the blocks have names and evidence. Duncker's functional fixedness, which prevents seeing the box of tacks as a support for the candle, and which is halved just by naming the problem differently. Luchins's Einstellung effect, which makes one apply the learned formula even when a simpler one exists. Tversky and Kahneman's anchoring, which in an ideation session is the first idea said out loud. Simon's satisficing, which stops the search at the first good-enough idea. Asch's conformity, which led 75 % of participants to give at least once an answer they knew to be false, and which drops to 5 % with a single dissenting voice. Janis's groupthink. The highest-paid person's opinion (HiPPO), which is hierarchy anchoring and conforming at the same time. And the preference for the known, which has an individual variant, status quo bias, and a group one that the authors of the How-Now-Wow matrix called the *creadox*: the group generates novel ideas and, when selecting, votes for the familiar ones.

Teresa Amabile adds the environment: norms of harshly criticizing new ideas, emphasis on the status quo, and excessive time pressure block; the interesting challenge, diverse teams, and freedom of execution stimulate. Each of those blocks has a remedy in the protocol of the next chapter.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why generation and evaluation are separated, and where the rule comes from.
- Assess an ideation session by fluency, flexibility, originality, and elaboration, counting categories and not notes.
- Recognize design fixation and fixation on one's own first idea, and explain why expertise does not immunize.

- Identify in a session anchoring, conformity, the highest-paid person's opinion, and the group that generates novelty and votes for the familiar.

Common mistakes and antipatterns

Measuring ideation by the number of notes. Eighty variations of three ideas are not eighty ideas; without counting categories, the team confuses fluency with variety.

Showing an example before ideating. The example fixates, even when its flaws are pointed out; if references must be shown, they come after the first individual round.

Trusting expertise. Experts fixate just the same and notice it less; the design professors in Linsey's study proved it.

Ignoring the environment. A group with a boss who criticizes new ideas or with ten minutes to ideate produces what it already knew.

Further reading

· Leahy et al. — Design fixation from initial examples

· Crilly — Fixation and creativity in concept development

· Bilalić, McLeod, and Gobet — Inflexibility of experts: reality or myth?

· Amabile — Componential theory of creativity

· Creative Education Foundation — What is CPS?

· Gamestorming — How-Now-Wow matrix

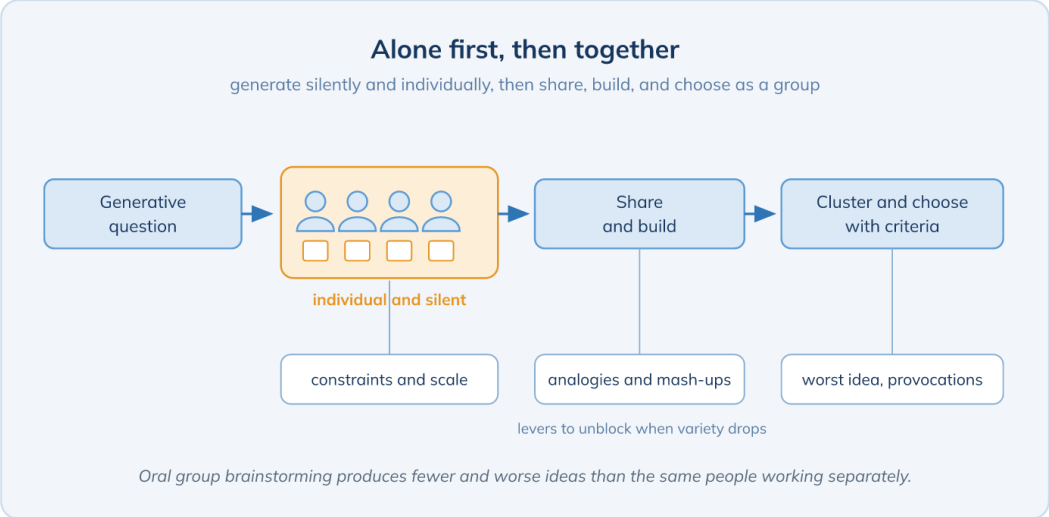
BLOCK D · DIVERGING WITH METHOD AND CONVERGING WITH CRITERIA

Chapter 11. The hybrid mode and the ideation protocol · ESTABLISHED

The evidence against oral brainstorming, the "alone first, then together" mode, the protocol common to all the catalogs with its levers, and the human-first order when AI comes in.

Introduction

If there is one matter on which popular practice and the evidence have disagreed for seventy years, it is group brainstorming. The evidence won, and the reference frameworks have incorporated it without saying so out loud: today nobody who knows how to facilitate asks ten people to shout ideas around a whiteboard. In this chapter we present that evidence, the way of working that follows from it, the protocol shared by all the method catalogs and the levers for unblocking a session, and we close with what changes when artificial intelligence comes in to ideate.



11.1 What the evidence says

Osborn promised in 1957 that an average person produces twice as many ideas in a group as alone. In 1958 Taylor, Berry, and Block tested it with the test research has used ever since, comparing a real group with a "nominal" group formed by adding up the ideas of people who worked separately, and found the opposite: the nominal groups produced almost twice as many. Diehl and Stroebe reviewed twenty-two experiments in 1987, eighteen in favor of the nominal groups, and isolated the main cause, **production blocking**: since only one person speaks at a time, the others forget or discard their ideas while they wait. The 1991 meta-analysis by Mullen, Johnson, and Salas quantified the loss, greater in quality than in quantity, and worse with large groups, with an authority present, and speaking instead of writing; their conclusion was that the popularity of brainstorming was "unequivocally mistaken."

Girotra, Terwiesch, and Ulrich closed the debate in 2010 by comparing two structures with the same people: the team that ideates together and the **hybrid mode**, which ideates first alone and then in a group. The hybrid tripled the ideas per unit of time, improved the average quality and that of the best idea, and revealed two more things. Groups are poor judges of their own ideas (their evaluation correlates less than 0.2 with that of external judges), and building on others' ideas, brainstorming's

favorite rule, produces neither more ideas nor better ones, and even reduces the average quality. Kohn and Smith added **collaborative fixation** in 2011: exposure to others' ideas before generating reduces the categories explored and conforms one's own, also when the exchange is by chat.

11.2 Alone first, then together

Practice has converged with the evidence. Jake Knapp writes in the Design Sprint guide that "group brainstorms don't work" and has each person sketch in silence; his Note-and-Vote meeting technique generalizes the pattern. The d.school starts every brainstorm with a generative question and individual time, and IDEO's seven rules (defer judgment, encourage wild ideas, build on the ideas of others, stay focused on the topic, one conversation at a time, be visual, go for quantity) govern the collective moment that comes afterwards. This skill's rule is summed up in four words: alone first, then together. Each person generates silently and in writing, with time and a quota; then ideas are shared without selling, built on and combined, and clustered. Building on others' ideas is a second phase, not the main source.

Write, don't talk. Almost everything the evidence recommends fits in one rule: that each person writes their ideas before anyone speaks. Rohrbach's brainwriting (six people, three ideas, five minutes, six rounds) formalized it in 1968 and digital boards make it easy.

11.3 The protocol and its levers

Crossing the catalogs of IDEO, the d.school, LUMA, IBM, the Design Sprint, and Gamestorming, a common protocol appears, more than a list of techniques. First, a well-formulated generative question, the "How might we…?" of chapter 9. Second, individual and silent generation with time and a quota; rapid sketching (the Design Sprint's Crazy 8s, eight sketches in eight minutes; IBM's storyboards) is the dominant way of doing it, because drawing forces one to be concrete. Third, a collective moment governed by IDEO's rules of conduct in which ideas are shared, built on, and combined. Fourth, levers to unblock when variety drops. And fifth, the clustering that prepares the convergence of the next chapter.

The most used levers are few. Deliberate constraints and change of scale ("what if it had to cost more than a million? and less than a euro?", "how would a field hospital do it?"). IDEO's analogies and mash-ups, which look in other contexts for the quality wanted and bring it over. The worst possible idea, which relieves anxiety and is inverted afterwards. Edward de Bono's provocations and random word. And bodystorming, which acts the idea out instead of describing it. The creativity-manual techniques, SCAMPER, the morphological matrix, TRIZ, or the lotus blossom, live in facilitation repositories and not in the Design Thinking frameworks; they are a catalog for reference, not the method. Co-creation with users, from chapter 17, is also an ideation technique, because they bring constraints and analogies the team does not have.

11.4 With AI

The evidence on ideation with artificial intelligence, accumulated between 2023 and 2026, repeats the pattern of groups: it raises the individual average and reduces the variety of the whole. The studies agree that AI-assisted ideas are rated better, especially those of the less creative people, and at the same time are more alike, and that AI performs better on incremental ideas than on radical

ones. The remedy is the same as with people: the individual human round comes first and the AI enters afterwards, as a lever to vary and combine, with prompts that force diversity (asking for step-by-step reasoning or different perspectives recovers part of the variety, according to Meincke, Mollick, and Terwiesch) and measuring categories afterwards, not ideas. Chapter 27 gathers the full evidence on homogenization.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why oral group brainstorming produces fewer and worse ideas than the same people separately, with the evidence that supports it.
- Facilitate an ideation in hybrid mode: individual and silent generation, sharing without selling, building, and clustering.
- Apply the common protocol (generative question, individual generation, collective moment with rules, levers, clustering) and choose the levers when variety drops.
- Bring AI into ideation after the human round, with prompts that force diversity and measuring categories.

Common mistakes and antipatterns

Starting by talking. The first idea said out loud anchors the whole group; the protocol starts with silence and writing.

Building on others as the main source. It is a valuable second phase for combining; as a starting point, it reduces variety and does not improve quality.

Applying SCAMPER to everything. The manual techniques are a reference repertoire; the protocol with four or five levers is enough for almost every session.

Asking the AI for ideas before the people. The team anchors on what the model proposed and the variety of the whole drops; the AI comes in afterwards and to vary.

Further reading

· Diehl and Stroebe — Productivity loss in brainstorming groups

· Mullen, Johnson, and Salas — Productivity loss in brainstorming groups: a meta-analytic integration

· Girotra, Terwiesch, and Ulrich — Idea generation and the quality of the best idea

· IDEO — Brainstorm rules

· Knapp — Note and vote

· LUMA — Creative matrix

· Meincke, Mollick, and Terwiesch — Prompting diverse ideas

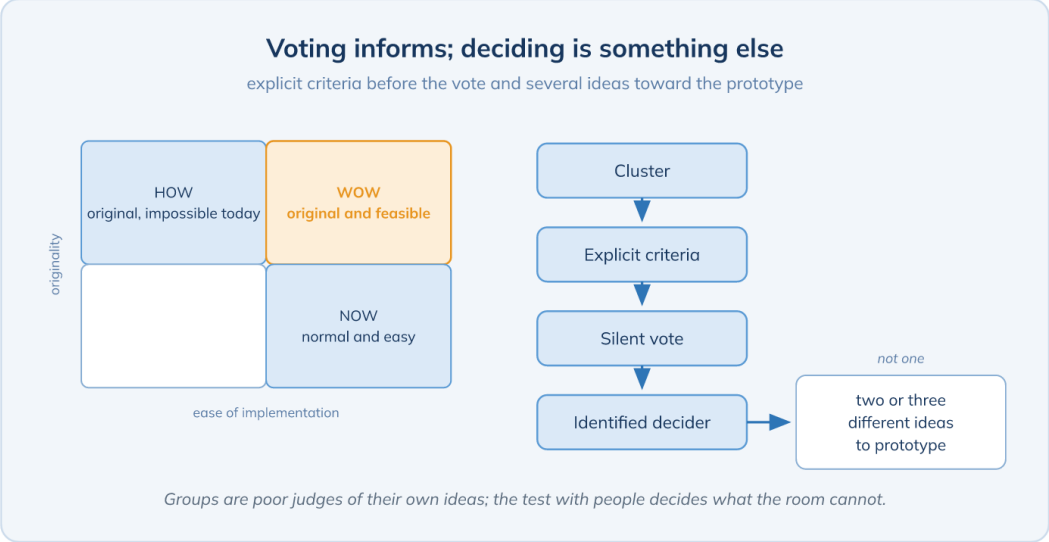
BLOCK D · DIVERGING WITH METHOD AND CONVERGING WITH CRITERIA

Chapter 12. Converging with criteria · ESTABLISHED

Clustering before voting, explicit criteria and matrices, the difference between voting and deciding, and why convergence ends in a prototype and not in the vote.

Introduction

Converging is the part of ideation that is taught least and decides most. A group that has generated sixty varied ideas can throw them all away in five minutes of badly run voting, keeping the most familiar one, the boss's, or the one that was best explained. In this chapter we see how to converge with criteria: clustering first, deciding with explicit criteria, understanding the biases of the vote and, above all, accepting that convergence does not end in the vote but in the prototype and the test.



12.1 Clustering before voting

The first step of convergence is clustering, and it has a reason NN/g calls the split vote: if five variants of the same idea compete separately, they share out the votes and lose to a single idea with less real support. Clustering by affinity (chapter 8), extracting the best of each group and turning it into concepts, which is what IDEO calls *bundle ideas*, prepares a selection among genuinely different alternatives. It is also the moment to count categories: if four families appear when clustering, the divergence session is not over.

12.2 Explicit criteria and matrices

The mature frameworks do not vote blind. The d.school's Bootleg asks for three voting criteria to be designated before voting ("the most likely to delight," "the rational choice," "the most unexpected") and one idea chosen per criterion, or for four categories to be kept, the rational one, the one that delights, the darling, and the long shot, so as not to lose variety when converging. Matrices make the criteria visible. Gamestorming's How-Now-Wow crosses originality with ease of implementation and was born to break the creadox, because it separates the original and feasible (wow) from the normal and easy (now) and from the original but impossible today (how). The NUF test scores each idea on new, useful, and feasible, first alone and then aloud, and its purpose is not to eliminate but to detect

where to improve them. LUMA's importance-difficulty matrix, IBM's prioritization grid, and IDEO's desirable, viable, and feasible triad serve the same function. A concept poster (for whom, what problem, how it works, why it is different) before voting avoids voting by headline.

12.3 Voting is not deciding

Dot voting is universal and its biases are described: the persuasion of whoever campaigns, the split vote, the bandwagon of voting for what already has dots, and hierarchy, when the boss votes first. The mitigations are simple: time to study the options, silent or anonymous voting, voting order from lowest to highest rank, weighted dots or dots per criterion in colors. The Design Sprint takes it further with its sequence of heat map (small dots on the interesting elements of each sketch, without discussion), a three-minute speed critique per sketch, a straw poll, and, at the end, the Decider's supervote, which may follow the group or not. The idea is that the group informs and one person decides, and the decision holds because it has an owner. Graded agreement scales, such as fist of five, measure dissent better than a yes or no.

Convergence ends in a prototype. Girotra and colleagues showed that groups are poor judges of their own ideas. That is why serious convergence is not settled with a vote: two or three different ideas are taken to prototype and the test with people is allowed to decide what the room cannot.

12.4 Two or three ideas, not one

The d.school states it as a rule: do not narrow too early, do not choose only one idea, and do not keep the safe options, because an implausible idea can trigger a useful finding and because taking two or three to prototype "preserves your innovation potential" compared with keeping the only one the majority accepts. IDEO U says it another way: the difference between the outrageous and the brilliant is not much. Group decision-making in general, with its rules of consent, gradients of agreement, and record, is covered by the Agile Inception skill; what is specific here is that in design the final decision is deferred to the learning of the next block.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Cluster ideas into concepts before voting and detect when divergence is not over.
- Set explicit criteria before selecting and use a matrix (How-Now-Wow, NUF, importance and difficulty) to make them visible.
- Facilitate a dot vote mitigating its biases and distinguish voting from deciding with an identified decider.
- Take two or three different ideas to prototype instead of a single one and explain why.

Common mistakes and antipatterns

Voting without clustering. The variants of an idea share out the votes and the one with no competition wins.

Voting without criteria. Without explicit criteria the group votes for the familiar, the boss's, or the best explained; the creadox has a name for a reason.

Taking the vote for a decision. The vote informs; if nobody has the power to decide and to hold the decision, it is reopened at the next meeting.

Keeping one idea. Choosing the only one everyone accepts is the shortest route to what the team was already doing.

Further reading

· NN/g — Dot voting

· Gamestorming — How-Now-Wow matrix

· Gamestorming — NUF test

· d.school — Design Thinking Bootleg (brainstorm selection)

· Knapp — Design Sprint guide (decide)

· Skill Arena — Agile Inception

BLOCK E · PROTOTYPING TO THINK AND TO LEARN

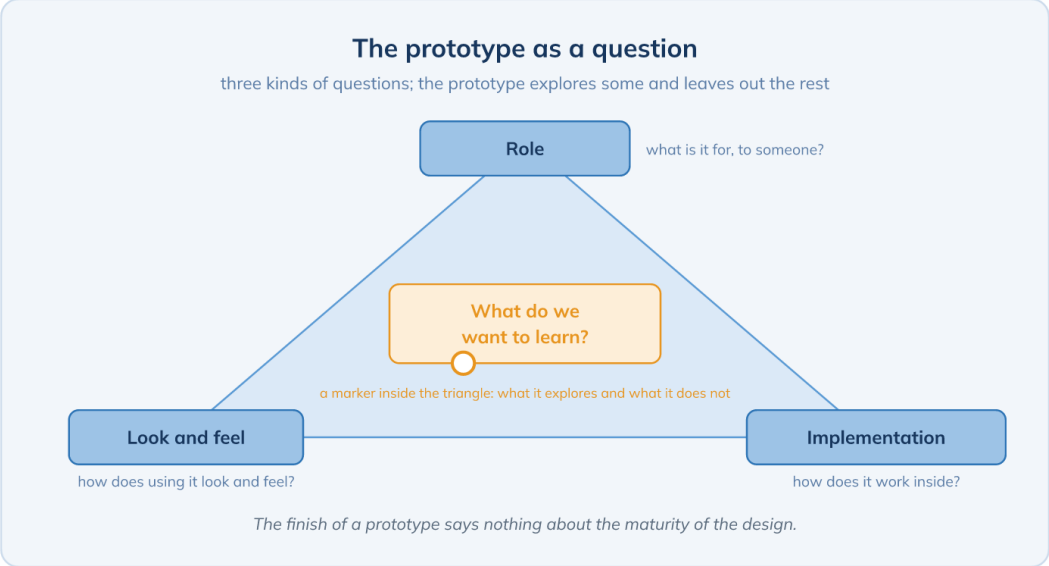
Chapter 13. What a prototype is and the prototype as a question ·

ESTABLISHED

What a prototype is and what it is for, the difference from the sketch, the three kinds of questions it answers, the dimensions of fidelity, and why finish says nothing about the maturity of the design.

Introduction

Product Owner training already teaches the ladder of evidence, from the cheap experiment to the gradual launch, and the difference between building to learn and building to earn. This block does not repeat it. What it adds is design's way of thinking about the prototype: that it is a question made tangible, that it serves to think and to converse and not only to validate, and that its fidelity is set by the question. In this chapter we fix those foundations with the model that has remained the reference since 1997 and with the experimental evidence that supports it.



13.1 What a prototype is and what it is for

Stephanie Houde and Charles Hill, then at Apple, gave in 1997 the broadest and most useful definition: a prototype is any representation of a design idea, in any medium, including an existing object when it is used to answer a design question; and a designer is anyone who creates a prototype in order to design, whatever their job title. Today that includes the Product Owner. Bill Buxton added in 2007 the distinction from the **sketch**: the sketch is quick, cheap, disposable, and plentiful, and it suggests and explores without committing; the prototype answers and refines. Much of what block D called making tangible were sketches in this sense.

What a prototype is for is summed up by IDEO U and the d.school. It is built to think (externalizing an idea and seeing it), to learn (testing it with people), and to communicate (winning support); and also, as the Bootleg reminds us, to empathize, because putting someone in front of an object reveals more than asking them, and to decide between alternatives. Tim Brown wrote in 2008 the sentence that holds the whole block together: the more finished a prototype seems, the less likely its creators

will be to pay attention to and profit from feedback; the goal of prototyping is not to finish, it is to learn.

13.2 The three kinds of questions

Houde and Hill's model is a triangle with three vertices, which are three kinds of questions a prototype can answer. **Role**: what part the artifact plays in someone's life, what it is for them. **Look and feel**: what concrete sensory experience using it produces. And **implementation**: with what techniques and components it works inside. A prototype sits within the triangle according to what it explores, and the authors insist that it is as important to say what it explores as what it leaves out; integration prototypes, which balance the three, are the most expensive and the last. The practical consequence is the question that opens all prototyping: what do we want to learn, what uncertainty do we want to reduce, and therefore what kind of prototype is needed. The d.school phrases it as identifying a variable and Houde and Hill themselves as shifting attention from the attributes of the prototype toward the questions about the design.

13.3 Just enough fidelity

Fidelity is not a single scale. Virzi, Sokolov, and Karis showed in 1996 that it is a continuum with several independent dimensions, breadth of features, degree of functionality, similarity of interaction, and aesthetic refinement; NN/g sums them up as interactivity, visuals, and content. The rule that follows is to raise fidelity only in the dimension the question needs: real content with rough visuals to test comprehension, faithful interaction with gray screens to test a flow. Houde and Hill also separate three things that get confused: **resolution** (how much detail), **fidelity** (how closely it resembles the final design), and the **maturity** of the design. They warn that finish does not indicate maturity: a polished model can be made very early for a market study and a rough one very late to test the structure.

The experimental evidence is solid and counterintuitive. Virzi and colleagues, and later Walker, Takayama, and Landay in 2002 with 28 participants and 169 usability problems, found that low- and high-fidelity prototypes detect the same usability problems. What low fidelity does not measure, according to Sauer and Sonderegger, are task times and aesthetic judgments. And early high fidelity has documented costs: designers resist changing what is already polished and explore the solution space less, stakeholders read the finish as "almost done," and Walker's review records the most human reason for raising fidelity too early, the fear of looking unprofessional.

Fidelity is set by the question. Not the calendar, not the client, not the tool. One decides what one wants to learn and builds the minimum that makes it visible, with high fidelity only in the dimension that question needs.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Define a prototype as any representation made to answer a question and distinguish it from the sketch.
- State the five purposes of prototyping (thinking, learning, communicating, empathizing, deciding) and apply the appropriate one.

- Place a prototype in the triangle of role, look and feel, and implementation, and make explicit what it explores and what it does not.
- Choose fidelity by dimensions according to the question and explain, with the evidence, why early high fidelity harms learning.

Common mistakes and antipatterns

Prototyping without a question. A prototype that does not know what it wants to learn tests everything and teaches nothing; the variable is identified before building.

Confusing finish with maturity. That the prototype looks finished says nothing about whether the design is, and it makes the feedback go quiet.

Raising all fidelity at once. It is enough to raise it in the dimension the question needs; the rest adds cost and anchors.

Taking the sketch for a prototype. The sketch explores and admits ambiguity; the prototype answers. Asking a sketch for prototype feedback kills it before its time.

Further reading

· Houde and Hill — What do prototypes prototype?

· Buxton — Sketching User Experiences

· Brown — Design Thinking (HBR, 2008)

· IDEO U — Introduction to human-centered prototyping

· Virzi, Sokolov, and Karis — low- and high-fidelity prototypes

· Walker, Takayama, and Landay — High-fidelity or low-fidelity, paper or computer?

· NN/g — UX prototypes: low fidelity vs. high fidelity

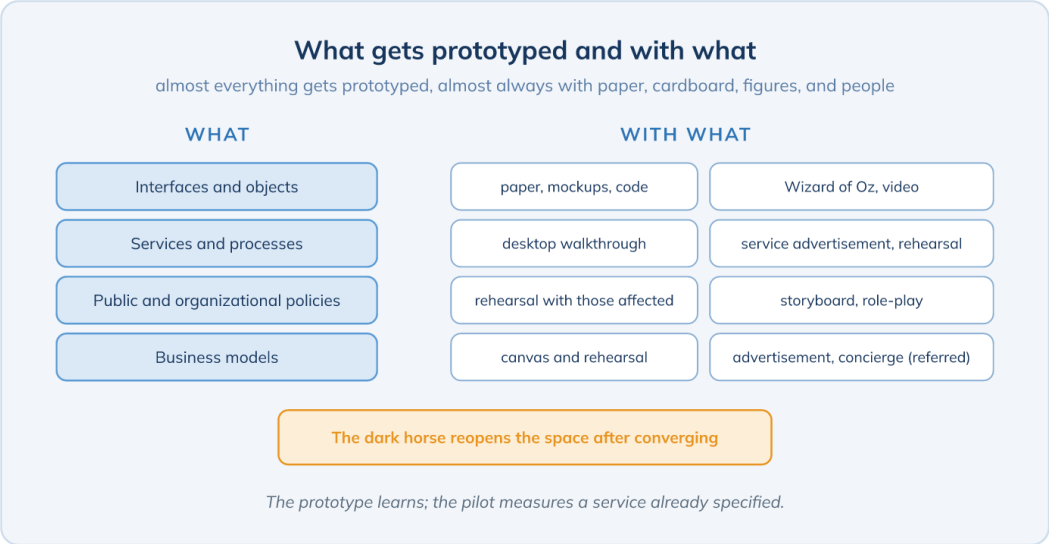
BLOCK E · PROTOTYPING TO THINK AND TO LEARN

Chapter 14. What gets prototyped and with which techniques · ESTABLISHED

Interfaces, objects, services, processes, policies, and business models; the prototype versus the pilot; and the techniques, from paper prototypes to the Wizard of Oz and conversational prototypes.

Introduction

When people think of prototypes they think of screens, and that is the most limiting idea a team can have. Services, processes, policies, and business models get prototyped, and the techniques for doing so are mostly paper, cardboard, figures, and people. In this chapter we go through what can be prototyped and with which technique, with the sequence of prototypes Stanford teaches for complex systems, the difference between prototype and pilot that the British public sector fixed, and an ethical warning about the techniques that fake. Demand-validation techniques, the fake door, the landing page, or the concierge, belong to discovery and to Product Owner training and are only named here.



14.1 What can be prototyped

Interfaces and software are the best documented case and the one the Discovery skill already covers with its chapter on prototype tests. Physical objects have their own discipline: Stanford's ME310 course teaches a sequence of prototypes for learning that starts with the **critical function prototype**, which builds and tests the most ambiguous or difficult element of the system before any other, and the **critical experience prototype**, which checks whether the concept adds value for someone, often with a Wizard of Oz; then come the dark horse, the funky system, the functional system, and the final. Services and processes are prototyped as lived sequences: the reference book on service design proposes the desktop walkthrough, an end-to-end simulation with toy figures on a map that is iterated in one or two hours, and the service advertisement, a poster that prototypes the value proposition and is tested with outsiders before investing in anything.

Public and organizational policies also get prototyped. Nesta formulated it in 2011 for public services as iterative, low-cost, low-risk experimentation, and the British government's Policy Lab has run more than 250 projects since 2014 with policy and service prototypes. For internal processes and an

organization's policies there is no school with a name, and at Scrum Manager we treat them as services: they are rehearsed with the people affected before being institutionalized. Business models are prototyped with the canvas and with the service rehearsals that accompany it.

Prototype and pilot. Nesta separates what many organizations confuse. The prototype develops, rehearses, or tests parts of a service or the whole idea in order to learn; the pilot measures the results of a service already specified in its final stage. Going to pilot without having prototyped is the expensive way of discovering that the idea did not work.

14.2 The techniques

The paper prototype, which Carolyn Snyder systematized in 2003, remains the cheapest way to test flows, terminology, and structure before programming; Jakob Nielsen observed that its brake is psychological, because "it feels like cheating." The storyboard draws the experience in sequence, and in the Design Sprint it is the blueprint of the prototype built the next day. Bodystorming and role-playing act the idea out in the real place and give immediate feedback. The **Wizard of Oz**, which John Kelley named in 1983, has a hidden person simulate the system's intelligence, and it is today the reference technique for prototyping conversational assistants and agents before building them, with its known limits (it does not scale, the wizard gets tired, the user may suspect). Click-through mockups test navigation; video prototypes a vision before it exists. ME310's **dark horse** deserves separate mention: after converging, a prototype of the risky, radical, or little-studied solution is built on purpose, to reopen the space and combat fixation, and its elements usually survive in the final concept.

The ethics of the techniques that fake is not a detail. Alberto Savoia, who systematized "pretotyping" at Google, warns against testing with a fake door a medicine that does not exist, and IxDF points out that the Wizard of Oz biases the results if the user suspects. Prototyping is faking in order to learn; one has to decide how far and say so afterwards.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Choose what to prototype (interface, object, service, process, policy, business model) according to the question and distinguish prototype from pilot.
- Apply the sequence of prototypes for learning about complex systems, starting with the critical function and the critical experience.
- Prototype a service or a process with a desktop walkthrough or a service advertisement.
- Choose the technique (paper, storyboard, bodystorming, Wizard of Oz, mockup, video, dark horse) and apply the ethical criterion to those that fake.

Common mistakes and antipatterns

Prototyping only screens. Services, processes, and policies are prototyped with figures, posters, and people; if the team only knows how to make mockups, it leaves out most of the problem.

Piloting without prototyping. The pilot measures a specified service; using it to find out whether the idea works costs what not finding out would cost.

Starting with the easy part. The critical function is the most ambiguous and difficult part of the system, and that is why it is prototyped first.

Faking without limit. A Wizard of Oz or a fake door on something that cannot be delivered deceives real people; the ethical criterion is decided beforehand.

Further reading

· Domingo et al. — Strategic prototyping to learn (ME310)

· This Is Service Design Doing — methods

· Nesta — Prototyping public services

· Nielsen — Paper prototyping

· Savoia — Pretotyping

· Skill Arena — Discovery and customer validation

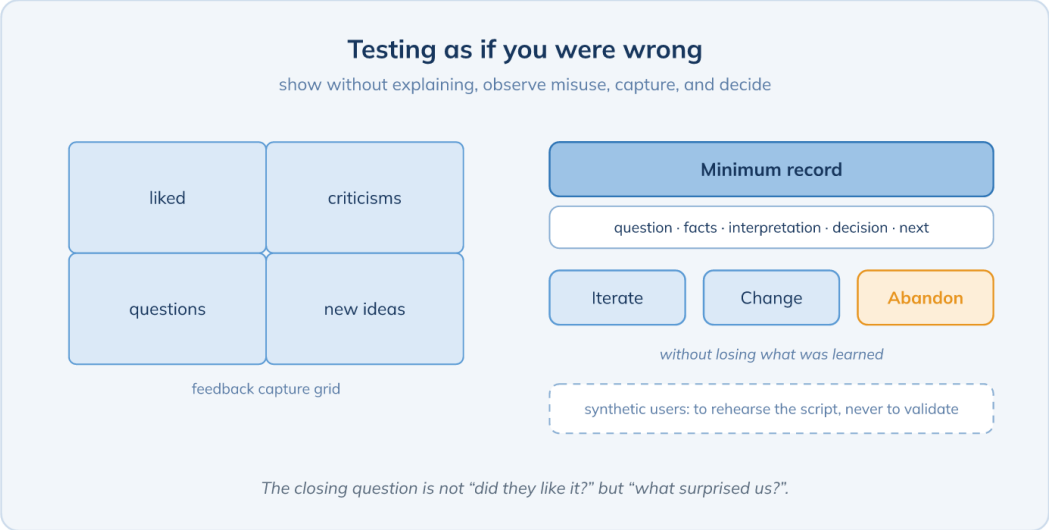
BLOCK E · PROTOTYPING TO THINK AND TO LEARN

Chapter 15. Testing in Design Thinking mode · ESTABLISHED

Showing before explaining, observing use and misuse, learning about the problem and not only about the solution, capturing feedback, recording and deciding, and what AI does not replace.

Introduction

Testing a prototype in design mode resembles a usability test and is something else. The usability test, which the Discovery skill teaches with its protocol of tasks, neutral moderator, and representative participants, measures whether a defined solution can be used. The test in Design Thinking mode seeks to learn about the problem, about people, and about the direction, and accepts that the most valuable thing it can teach is that the problem was framed wrongly. In this chapter we see how it is conducted, how what is observed is captured, how one decides afterwards, and why artificial intelligence does not replace people in this step.



15.1 Showing before explaining

The d.school Bootleg's testing card sums up the protocol in four gestures. Show, don't tell: the prototype is put in the user's hands, or the user inside the prototype, with the minimum context to know what to do, and they are asked to think aloud. Observe actively: the user is not corrected when they "get it wrong," because misuse is the information. Ask afterwards, which is usually the most valuable part ("show me why this would or wouldn't work for you"). And answer questions with questions ("what do you think that button does?"). The Design Sprint adds the Friday rule: five interviews in one day, the whole team watching from another room and taking notes, and the patterns appear from the third onward. Jakob Nielsen gave the basis in 2000: five users uncover most of the problems, and three rounds of five yield more than one of fifteen because between rounds the design changes.

15.2 Learning, not validating

Confirmation bias, which Raymond Nickerson defined as seeking or interpreting evidence in favor of what one already believes, takes a concrete form in prototyping: prototyping to prove the idea was

good. The insidious thing is that it is almost never deliberate; the team builds its case without knowing it, chooses the participants who agree with it, and reads ambiguous reactions as approval. The d.school turned it into a rule: prototype as if you know you're right, but test as if you know you're wrong. And it adds that the test also serves to gain empathy, because observing someone with an object reveals what an interview does not draw out, and to revise the point of view, because the prototype can show that the solution was wrong and also that the problem was.

Building to learn. A team that seeks validation hears what it wants; one that seeks learning writes down what it did not expect. The closing question of each test is not "did they like it?" but "what surprised us?".

15.3 Capturing, recording, and deciding

The capture formats order what is observed in the moment. The d.school's **feedback capture grid** has four quadrants: what was liked, the criticisms, the questions that arose, and the new ideas. "I like, I wish, what if" serves both for user feedback and for the team's own. LUMA's **rose, thorn, bud** technique codes each observation as positive, negative, or with potential, with stickers in three colors, and although LUMA presents it for analyzing data in general, in practice it is used when closing a round of tests. IDEO proposes a results sheet per prototype.

What the skill fixes as a minimum record fits in five lines: what question the prototype asked, what was observed (facts and quotes), what was interpreted, what was decided, and what prototype comes next. The possible decisions are three. Iterate, when the direction is good and adjustments are needed. Change direction, when the test shows that the problem or the solution was framed wrongly. And abandon, which Jeff Patton describes from the dual track as killing unviable ideas early, "ideally in hours or days," without losing what was learned; ME310's dark horses are almost always discarded and their elements survive in the final concept. The distinction between fact, quote, and interpretation in the notes is the one the Discovery skill teaches.

15.4 With AI

Synthetic users, profiles generated by a language model that are "interviewed" or made to react to a prototype, are the recent temptation of this step. Nielsen Norman Group compared them in 2024 with three real studies and found that they are complacent (they tend to please and to approve of any idea), they overpromise (they complete courses that real users abandoned), they produce generic lists without prioritizing, and they provide no behavioral data. They serve to rehearse a test script, anticipate objections, and generate hypotheses; they do not serve to validate a concept or to present their answers as research, and the Product Owner 2.0 guide already calls that research theater. A conversational prototype generated with the model itself is a legitimate prototype of the service; a user simulated by the model is not a user. Chapter 27 gathers the evidence on what those profiles distort.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Conduct a prototype test in design mode: show without explaining, observe use and misuse, ask afterwards, and answer with questions.

- Recognize confirmation bias in its form of prototyping to prove and design the test to learn.
- Capture feedback with a grid or with rose, thorn, bud, and record question, facts, interpretation, decision, and next prototype.
- Decide between iterating, changing direction, or abandoning without losing what was learned, and not replace people with synthetic users.

Common mistakes and antipatterns

Explaining the prototype before showing it. The user who already knows what to do does not reveal how they would understand it on their own.

Correcting the user. Misuse is the data; interrupting it destroys it.

Asking whether they liked it. Approval teaches nothing; what surprised does.

Validating with synthetic users. They approve of everything and do not behave; presenting their answers as research is manufacturing evidence.

Further reading

· d.school — Design Thinking Bootleg (testing with users, feedback capture)

· LUMA — Rose, thorn, bud

· Nielsen — Why you only need to test with 5 users

· Nickerson — Confirmation bias

· Patton — Dual track development is not duel track

· NN/g — Synthetic users

· Skill Arena — Discovery and customer validation

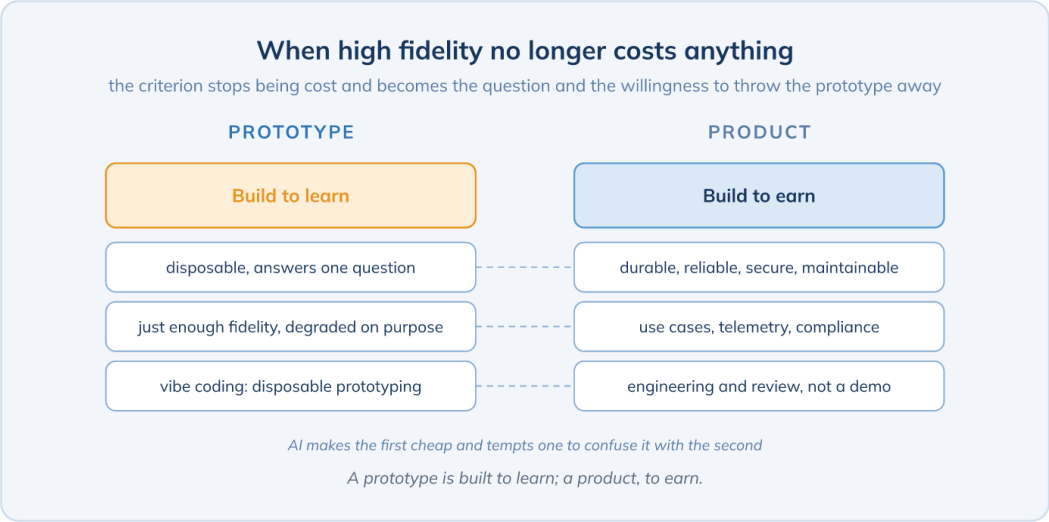
BLOCK E · PROTOTYPING TO THINK AND TO LEARN

Chapter 16. Prototyping with artificial intelligence · EMERGING

What has changed between 2023 and 2026, the criterion when high fidelity no longer costs anything, the line between prototype and product, the place of vibe coding, and the observed risks.

Introduction

In two years the functional prototype has gone from costing weeks of engineering to being generated by describing it in minutes, and whoever generates it is no longer only design or engineering. It is the biggest change prototyping has seen since paper, and the just-enough-fidelity doctrine of chapter 13 has to be reread in its light. This chapter gathers the chronology, what the evidence and the institutions say about using these tools, the line between prototype and product that has become the central discipline, and the observed risks. The Product Owner 2.0 guide and the Scrum in AI-enabled teams skill already cover the economics of the prototype and the fast development lane; what matters here is what changes in design work.



16.1 What has changed

The chronology is short. In October 2023 the first tool appeared that turned a description into an interface with code; in 2024 conversational assistants started showing executable artifacts alongside the chat and agents appeared that take an application from idea to deployment; in February 2025 Andrej Karpathy coined "vibe coding" to name programming by giving in to intuition and forgetting the code exists, for weekend projects; in May 2025 Figma announced its prompt-to-app tool and in July opened it to all plans; in November 2025 Marty Cagan wrote the reference text on prototypes and products; and in April 2026 Karpathy himself withdrew the term, leaving it for prototypes and personal tools and proposing "agentic engineering" for everything else. In that time, product managers and designers have gone from asking for prototypes to making them, and most of designers' AI use is still, according to industry surveys, in writing and documentation rather than visuals.

16.2 When high fidelity no longer costs anything

The classic doctrine said that early high fidelity was expensive and that is why it was avoided. Now it is free, and the criterion has to change its basis: it is no longer cost, but the question and the willingness to throw the prototype away. A functional prototype generated in ten minutes anchors the team as much as one that cost three weeks, and polished it looks just as finished; the difference is that nobody has an excuse not to throw it away. From there comes a new discipline, which IDEO U already teaches in its AI prototyping course as "rethinking fidelity" and "adding intentional friction": degrading fidelity on purpose when the question is about the concept, so that people react to the idea and not to the finish, and using the speed to explore more directions instead of to polish one. Nielsen Norman Group evaluated nine tools in 2025 by redesigning a real page and found that they produce surfaces that look final without the logic or the usability underneath, that they do not weigh design trade-offs, and that they lean toward dominant patterns, with "flat and interchangeable" designs; its recommendation is to use them for early conceptual exploration, proofs of concept, and quick prototypes for usability tests, in the hands of someone who understands the craft.

The question and the wastebasket. When high fidelity costs nothing, a good AI prototype is recognized by two things: it knows what question it answers and it is thrown away without pain once it has answered it.

16.3 Prototype and product

Cagan formulated it with the clarity that was needed: prototypes are built to learn and products to earn, and a product faces dozens or hundreds of use cases, complex business logic, reliability, telemetry, scaling, security, integrations, and recovery, that is, everything the prototype avoids. The tools have forked, some for prototyping and others for building product, and some vendors promise far more than they deliver. The risk Cagan has seen first-hand is the product manager who shows a functional prototype with real data and creates in the organization the expectation that "it's almost done"; Thoughtworks has it on Hold in its radar as AI-accelerated shadow IT, the new generation of the spreadsheets that hold up critical processes, and recommends distinguishing the disposable from the durable. Vibe coding is, by its author's definition, disposable prototyping; the Scrum in AI-enabled teams skill treats it as a fast lane separate from the production one.

16.4 The observed risks

The risks already have a name and a source, though little accumulated evidence. Anchoring through instant high fidelity, which fixes the team on the first generated solution. Generic and interchangeable patterns, which make all prototypes look alike and keep the rare solutions from appearing. The surface that looks final without logic or usability underneath, which deceives stakeholders and sometimes the team itself. Complacency with what is generated, which Thoughtworks documents as automation bias with code and which carries over to design. Ungoverned parallel systems. And a new legal obligation: a conversational prototype that is tested with people outside the lab falls under the transparency duty of the European AI Act, in force since August 2026, which requires informing them that they are interacting with a machine.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain what has changed in prototyping between 2023 and 2026 and who prototypes now.
- Decide the fidelity of an AI-generated prototype by the question and the willingness to throw it away, and degrade it on purpose when the question is about the concept.
- Distinguish prototype from product before the organization and prevent a functional prototype from creating delivery expectations.
- Recognize the observed risks (anchoring, generic patterns, surface without substance, complacency, parallel systems) and the transparency duty of conversational prototypes.

Common mistakes and antipatterns

Polishing because it is free. Instant high fidelity anchors just as much as the expensive kind; if the question is about the concept, it is degraded on purpose.

Showing the prototype as if it were the product. With real data and a final finish, the organization understands that it is done; the gap with the product is explained before showing it.

Accepting the first result. The tools lean toward dominant patterns; exploring several directions is what the speed allows and what almost nobody does.

Testing an agent without notice. A conversational prototype with people outside the lab must declare that it is a machine; it has been law since August 2026.

Further reading

· Cagan — Prototypes vs products

· NN/g — AI prototyping in real design contexts

· Thoughtworks — AI-accelerated shadow IT

· Figma — Figma Make general availability

· Willison — Not all AI-assisted programming is vibe coding

· IDEO U — Prototyping with AI

· Skill Arena — Scrum in AI-enabled teams

BLOCK F · CO-CREATING, FACILITATING, AND COMMUNICATING DESIGN WORK

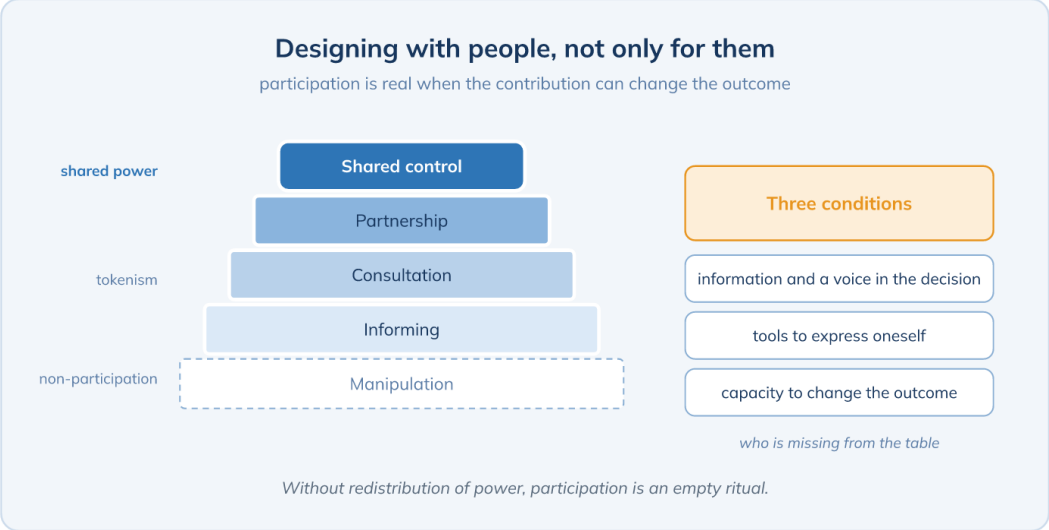
Chapter 17. Designing with: co-designers and the limits of participation ·

ESTABLISHED

Half a century of participatory design, the user as partner, the three conditions of real participation, and the limits that turn it into decoration.

Introduction

"Designing with people, not only for them" is one of the most repeated phrases of Design Thinking and one of the least honored. Honoring it has concrete conditions that the participatory design tradition has been documenting for fifty years, and failing at them produces what the literature calls tokenism and participation-washing. In this chapter we go through that tradition, the conditions of real participation, its documented limits, and the practical question that sums it all up: who is missing from the table.



17.1 Two genealogies and a definition

People's participation in design was born twice in the seventies. At the Design Research Society's Design Participation conference, in Manchester in 1971, whose preface, written by Nigel Cross, accused professional designers of having failed in their responsibility to anticipate the adverse effects of their projects. And in the Scandinavian trade union projects, from the Norwegian metalworkers' union in 1970 to the UTOPIA project with the Nordic graphic workers in 1981, which designed computer tools with the workers through mock-ups and "design by doing," because what was at stake was who controlled the work. Elizabeth Sanders and Pieter Jan Stappers systematized in 2008 the difference between the two traditions that converge in Design Thinking: the American one, user-centered, which treats the user as a **subject** who is observed and interviewed, and the Nordic, participatory one, which treats them as a **partner**. They defined co-creation as any act of collective creativity and co-design as that creativity applied to the whole process, with people without design training acting as experts of their own experience.

17.2 The three conditions of real participation

From that tradition come three conditions that can be checked. The first, which Finn Kensing fixed for the Scandinavian workers, is that people have access to the relevant information, the possibility of forming their own position, and some role in the decision. The second, from Sanders and Stappers, is that they have tools to express themselves, because nobody can act as an expert of their experience if they are only asked to talk: mock-ups, kits of parts, a cardboard frame so that shy people can tell their story. The third is Sherry Arnstein's in her 1969 ladder of citizen participation: that their contribution can change the outcome, because participation without redistribution of power is "an empty and frustrating ritual." The middle rungs of the ladder, informing, consulting, and placation, are tokenism: people hear and are heard, and have no power to make anyone heed them.

Who is missing from the table. Before a design session, the cheapest and most forgotten question: who is going to live with the outcome and is not here, and what would it take for their contribution to be able to change it?

17.3 The documented limits

Recent evidence shows that applying a participatory methodology does not guarantee participation. Mona Sloane and colleagues warned in 2020 of **participation-washing**, participation used as in international development, to overcome local resistance, and distinguished three forms: as unrecognized work, as one-off consultation, and as long-term justice. A 2021 study on co-design in healthcare documented what happens inside formally participatory processes: patients felt inferior ("of course, they were the experts and I wasn't"), professionals dismissed lived experience as anecdotal, and the roles of expert and novice were reproduced. Lucy Kimbell pointed out from academia that Design Thinking privileges the designer as the main agent, and Natasha Iskander's critique turns them into an interpreter of other people's needs.

The most radical answer is that of the Design Justice Network, founded in Detroit in 2015 and with principles published in 2018: center the voices of those directly affected, prioritize the impact on the community over the designer's intentions, see the designer as facilitator rather than expert, and look first for what already works in the community before designing anything new. IDEO, in its co-creation session card, puts it operationally: treat participants as designers, not as interview subjects. And there is a nuance worth knowing: the Scandinavian tradition sought consensus, while other participatory currents value dissent and friction; a design session does not always have to end in agreement.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Distinguish the user as subject from the user as partner and explain where each tradition comes from.
- Check the three conditions of real participation (information and a voice in the decision, tools to express oneself, capacity to change the outcome) in a specific session.
- Recognize tokenism, participation-washing, and professional hierarchies within a co-design process.

- Ask who is missing from the table and design the session so that their contribution can change the outcome.

Common mistakes and antipatterns

Calling consultation co-creation. If people's contribution cannot change the outcome, it is tokenism however many sticky notes there are.

Inviting without tools. Asking someone to talk about their experience without giving them something to express it with reproduces the role of subject.

Dismissing experience as anecdote. It is the usual way professionals regain power inside a co-design; lived experience is the data.

Always seeking consensus. Sometimes the session must make disagreement visible, not resolve it.

Further reading

· Sanders and Stappers — Co-creation and the new landscapes of design

· Arnstein — A ladder of citizen participation

· Sloane et al. — Participation is not a design fix

· Lindblom et al. — participation in a co-design process

· Design Justice Network — Principles

· IDEO — Co-creation session

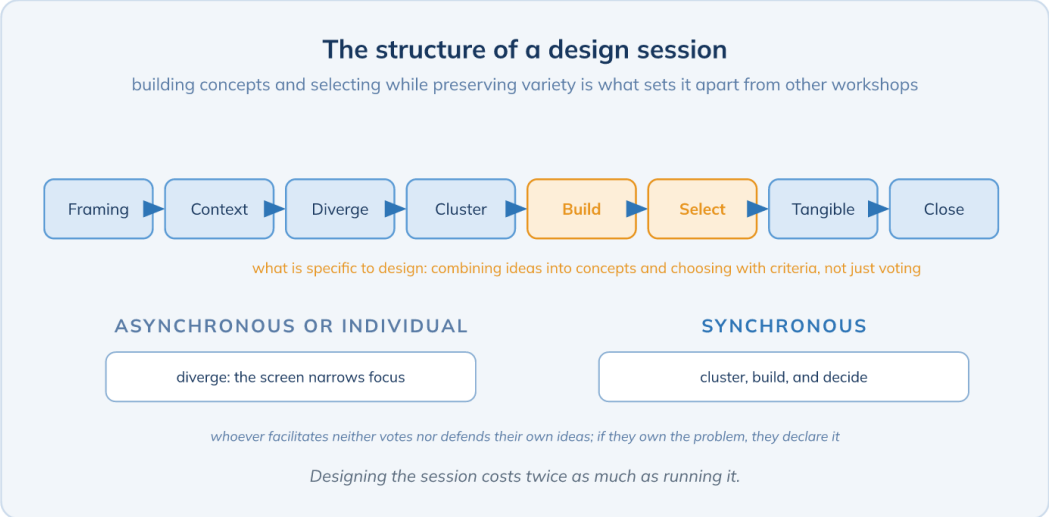
BLOCK F · CO-CREATING, FACILITATING, AND COMMUNICATING DESIGN WORK

Chapter 18. The design session, in the room and remote · ESTABLISHED

How a design session is designed, its archetypal structure, what distinguishes it from other workshops, the tension between facilitating and designing, and what is done on screen and what off it.

Introduction

Workshop facilitation is a discipline in its own right with its body of knowledge, and Scrum Manager's Agile Inception skill teaches it: content neutrality, dynamics, participation and power, group decisions, remote and hybrid. We do not repeat it here. What this chapter adds is what is specific to a design session: its archetypal structure, the phases other workshops do not have, the tension of someone who facilitates and designs at the same time, and the only experimental evidence on which part of creative work suffers in a video call.



18.1 Designing the session

A design session is designed, with the same elements as any workshop: an objective, a working question (the "How might we…?" of chapter 9), the participants, the sequence, the materials, and the time. Hyper Island's template calls them intention, desired outcomes, agenda, roles and rules, and time. Two practical rules from SessionLab: designing costs twice as much as delivering (a two-hour session requires about four hours of design) and it is wise to mark in advance which activities can be skipped if time runs short and to leave a contingency slot, because the agenda is a scaffold and not a script. Luke Hohmann, in his innovation games, set in 2006 the precedent of choosing the activity according to the objective (understanding needs, usage patterns, requirements, or future direction), which is exactly what block H calls designing your own process.

18.2 The archetypal structure and what is specific to design

Crossing the d.school, IDEO, LUMA, Nesta, and Hyper Island, the same structure always appears. Framing (intention, question, rules, decision authority). A brief warm-up related to the challenge. Loading the context, pouring what is known onto notes, in silence first and in a round afterwards.

Diverging, in the hybrid mode of chapter 11. Clustering by affinity. Building, combining ideas into concepts. Selecting while preserving variety, with criteria and not only with votes. Making tangible, with a sketch, a storyboard, or a prototype within the session itself. And closing with structured feedback, decisions, owners, and next steps.

What distinguishes this structure from an inception or a retrospective are three things. The building phase, in which ideas are combined and evolve into concepts with substance, and which the decision formats do not have. The selection that is not reduced to voting, with the d.school's four categories or its "bingo" (an idea that inspires a physical prototype, another a digital one, and another an experience one). And the ideation levers, the prepared variations of the statement and the constraints that the facilitator brings out when energy drops. Durations are not fixed in the literature; IDEO gives one to three hours for a co-creation session and sixty to ninety minutes for clustering and combining, and they are conventions.

18.3 Facilitating and designing at the same time

Classic facilitation demands neutrality about content, and whoever facilitates a design session almost never has it, because they are usually a designer or the owner of the problem. The literature resolves the tension in three ways. Separating roles, like the Design Sprint, with a Facilitator who runs the session and a Decider who decides. Declaring them, like Sanders and Stappers, who accept that in co-design roles mix and that the designer "still plays a critical role in giving form to the ideas," on condition of saying when one speaks as facilitator and when as designer. Or inverting them, like the Design Justice Network, which sees the designer as facilitator and not as expert. This skill's practical rule: whoever facilitates neither votes nor defends their own ideas during divergence, and if they are also the owner of the problem, the decision authority is agreed before starting.

18.4 Remote, and with AI

The only specific experimental evidence on remote creativity is that of Melanie Brucks and Jonathan Levav in *Nature* in 2022, with a laboratory study and a field experiment in five countries: video conferencing reduces the generation of creative ideas, because it focuses people on the screen and narrows their cognitive focus, and by contrast it does not hurt idea selection, which may even improve. The consequence for session design is direct: diverge asynchronously or individually, with time and without a camera, and reserve the video call for clustering, building, and deciding. Collaborative boards help with specific features, such as Mural's private mode, which hides individual contributions until everyone has finished and is the digital version of writing in silence, and with templates of the design methods (Miro has more than one hundred and eighty). The rest of the rules for remote and hybrid are in Agile Inception.

With artificial intelligence, the design session admits the same as any workshop, preparing the agenda and materials, proposing clusters that the group reviews, and drafting the minutes that a person corrects, and it has a warning of its own: clustering is interpretation, and if the AI clusters on the group's behalf, the group has not synthesized. The HPI in Potsdam gave a name in 2026 to the risk of the AI ending up leading the session without anyone having decided it, agency drift, and its answer is that of chapter 26: deciding at each moment who leads.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Design a design session with objective, working question, participants, sequence, materials, and time, marking what can be dropped.
- Apply the archetypal structure and its phases specific to design: building concepts and selecting while preserving variety.
- Declare or separate roles when whoever facilitates also designs or owns the problem.
- Distribute remote work according to the evidence: diverge off screen, converge on it; and use AI to prepare and propose without letting it cluster on the group's behalf.

Common mistakes and antipatterns

Improvising the session. Designing it costs twice as much as delivering it; without sequence or timings, the session ends in divergence and decides nothing.

Skipping the building phase. Going from notes to the vote leaves ideas as headlines; combining and evolving is what turns them into concepts.

Facilitating and defending one's own idea. Whoever facilitates neither votes nor sells during divergence; if they own the problem, they declare it and agree beforehand who decides.

Ideating by video call. The evidence says the screen narrows focus; diverge asynchronously and converge on the call.

Further reading

· Nesta and IDEO — Designing for public services

· Hohmann — Choosing Innovation Games to meet your goals

· SessionLab — Workshop design

· Brucks and Levav — Virtual communication curbs creative idea generation

· Mural — facilitation features

· Skill Arena — Agile Inception

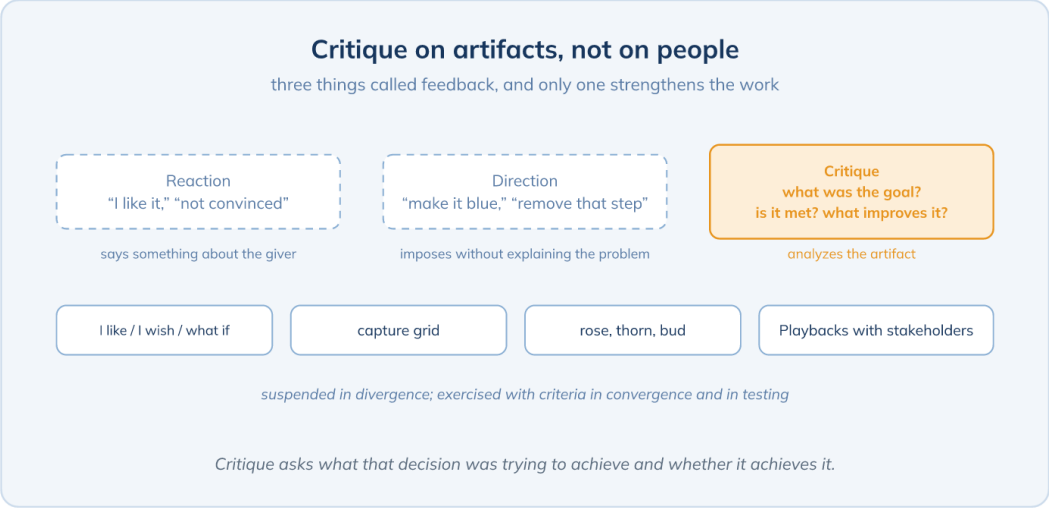
BLOCK F · CO-CREATING, FACILITATING, AND COMMUNICATING DESIGN WORK

Chapter 19. Critique and feedback on ideas and prototypes · CONSOLIDATING

Critique as a design practice versus feedback of reaction or of imposition, on artifacts and not on people, with its formats and safety to create as its condition.

Introduction

When Natasha Jen said that Design Thinking lacked peer critique she was right, and it is one of the best-founded criticisms of the approach. In any design school the crit is the heart of the training: the work is hung on the wall and analyzed with rigor and without mercy, and it comes out better for it. Popular Design Thinking replaced that with sticky notes and the vote. This chapter recovers critique as a practice, with its formats, its rules, and its condition, which is safety to create. The Communication and conflict management skill teaches interpersonal feedback; here critique is about artifacts.



19.1 Three kinds of feedback

Adam Connor and Aaron Irizarry, in the reference book on design critique, distinguish three things that go by the same name. **Reaction** feedback ("I like it," "it doesn't convince me") says something about whoever gives it and little about the work. **Direction** feedback ("make it blue," "remove that step") imposes a solution without explaining the problem. And **critique** analyzes whether the design meets its objectives and why, with questions such as what the objective of this decision is, whether it achieves it, and what would make it achieve it better. The authors observe that real critique "has become a lost skill" in teams and that it is often used to assert authority or push agendas under the guise of feedback. Critique is exercised on artifacts, ideas, sketches, and prototypes, and never on people; and it has its moment: it is suspended during divergence, where the rule is to defer judgment, and exercised with criteria in convergence and in testing.

19.2 The formats

The formats make critique practicable for teams that do not come from a design school. The d.school's "I like, I wish, what if," which its Bootleg calls "almost too simple to write down and too useful not to mention," structures feedback in first-person sentences and brief headlines, and serves

both for a prototype and for the session itself. The feedback capture grid orders in four quadrants what was liked, the criticisms, the questions, and the new ideas. LUMA's rose, thorn, bud technique codes each observation as positive, negative, or with potential, and forces one not to propose solutions while critiquing. And IBM's **Playbacks** institutionalize critique: periodic meetings with users and stakeholders in which the team tells the user's story with what it has built, which reveal misalignments and measure progress against the problem, not against the plan.

Critique strengthens, reaction flatters. A team that only receives "I like it" learns nothing; one that receives "do it this way" does not either. Critique asks what that decision was trying to achieve and whether it achieves it.

19.3 Safety to create and visible disagreement

Critique requires a condition that is usually named with a term borrowed from another field, psychological safety. It is worth knowing that the concept was born in the fifties with Carl Rogers, tied to fostering creativity, the unconditional acceptance that allows one to take risks, and that only later did Amy Edmondson take it to work teams, where meta-analyses link it to innovation and to people speaking up. In a design session, safety is not a matter of climate; it is the condition for someone to hang a bad sketch on the wall and for someone else to tell them why it does not work without anyone breaking. Hierarchy destroys it fast: the boss critiques first and the rest fall silent. And there is a nuance the participatory tradition recalls: the goal of a session is not always to converge on a shared idea; sometimes it is to make disagreement visible and record it, because a hidden disagreement reappears as a blockage.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Distinguish critique from reaction feedback and from direction feedback, and practice it on artifacts and not on people.
- Apply the formats (I like, I wish, what if; capture grid; rose, thorn, bud; Playbacks) at the right moment.
- Suspend judgment in divergence and exercise it with criteria in convergence and in testing.
- Create the conditions for a team to hang up imperfect work and critique it, and make disagreement visible when appropriate.

Common mistakes and antipatterns

Confusing critique with reaction. "I like it" and "it doesn't convince me" say nothing about the work; critique asks about the objective and whether it is met.

Critiquing during divergence. Judgment is deferred while ideas are being generated; critique has its moment afterwards.

Critiquing the person. Critique is about the sketch, the prototype, or the idea; about people it is something else and another skill covers it.

Resolving every disagreement. Some are better recorded and left visible; forcing consensus hides them and they come back.

Further reading

· Connor and Irizarry — Discussing Design

· d.school — Design Thinking Bootleg (I like, I wish, what if; feedback capture)

· LUMA — Rose, thorn, bud

· IBM — Enterprise Design Thinking framework (Playbacks)

· Skill Arena — Communication and conflict management

BLOCK F · CO-CREATING, FACILITATING, AND COMMUNICATING DESIGN WORK

Chapter 20. Visual thinking and communicating findings and prototypes

· ESTABLISHED

Why drawing is thinking in another way according to design theory, objects as a language for those who have no design language, and communicating what was learned deliberately to those who decide.

Introduction

"Be visual" is one of IDEO's seven brainstorming rules and a principle of the Design Council, and it is almost never explained why. This chapter explains it from design theory, which is where the explanation is solid, and not from pop neuroscience, which is not. And it adds the other half of communication in design, the one the most repeated criticism of the approach finds missing: telling what was learned and showing the prototypes to those who decide, in a way that informs, inspires, and does not replace the evidence.

20.1 Why drawing is thinking in another way

Nigel Cross explained it in 1982 in an article that is still the basis of design research. Designers learn to think "in this sketch-like form," in which the abstract patterns of the requirements are turned into the concrete patterns of an object; it is like learning an artificial language, a non-verbal code that translates messages in both directions between abstract requirements and concrete objects, and that facilitates constructive, solution-focused thinking, just as verbal and numerical codes facilitate analytical thinking. Bill Buxton added the second reason: the sketch must be ambiguous, because its expressive power lies in ambiguity, which invites interpreting and proposing instead of approving. And David Sibbet and Geoff Ball, from graphic facilitation, the third: a shared image is the group's explicit memory, what allows twenty people to talk about the same thing.

What should not be used as justification is pop neuroscience. Dual coding theory and the picture superiority effect are evidence about recognition memory and recall, not about idea generation or shared understanding, and their mechanism is disputed; phrases such as "the brain processes images sixty thousand times faster" have no source. Design's reasons are enough.

20.2 Objects as a language

Sketches, maps, diagrams, storyboards, and objects give a language to those who have no design language, and that is their function in the co-creation of chapter 17: Sanders speaks of tools for people to express themselves, and the Scandinavian projects used mock-ups so that workers could show a practical understanding they did not know how to put into words. The user journey map that the Scrum Manager BoK records is at once a synthesis tool and a conversation tool. Dan Roam summed up the method in four steps, look, see, imagine, and show, and Mike Rohde insisted on what IDEO already said, that no artistic skill is needed: ideas, not art. Graphic facilitation, which visually records what a group says while it says it, has existed since the seventies and remains the fastest way for a group to see that it has been understood.

20.3 Communicating findings and prototypes

The d.school includes communicating deliberately among its eight design abilities, IDEO U devotes an advanced course to storytelling for influence, IBM lists it as a competence of its Practitioner, and the Systemic Design Framework has a role called leader and storyteller. The reason is the most repeated criticism of the approach: the findings do not reach those who decide. Communicating what was learned consists of narrating the research with stories, quotes, and surprises, so that it informs and inspires at the same time; of showing before explaining also to those who decide, because a prototype in the hands convinces more than a presentation; of adapting the story to the audience without losing the evidence; and of using the prototype as an argument while knowing that a brilliant demo can decide on its own what should be decided with data. Tim Brown, responding to the critique in 2023, called storytelling the key activity of change; it is worth taking seriously and with caution.

Show it to those who decide too. The same rule as the test with users holds before a committee: a prototype that can be touched and a story with people's voices convince more than twenty slides. The risk is that they convince too much.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why drawing is thinking in another way with the reasons from design theory, without resorting to pop neuroscience.
- Use sketches, maps, storyboards, and objects to give a language to those who have no design language.
- Narrate the findings of a research effort with stories, quotes, and surprises, adapting the story to the audience without losing the evidence.
- Show a prototype to those who decide and recognize when a demo is deciding on its own.

Common mistakes and antipatterns

Justifying the visual with neuroscience. The figures in circulation have no source; design's reasons are better and are true.

Demanding artistic skill. Ideas, not art; a clumsy sketch that is understood is worth more than a pretty one that does not communicate.

Communicating only with slides. Findings that do not reach those who decide are the reason for the most repeated criticism of the approach; a story and a prototype do reach them.

Letting the demo decide. A brilliant demo convinces too much; the evidence that accompanies it is what the organization must see.

Further reading

· Cross — Designerly ways of knowing

· Buxton — Sketching User Experiences

· Sibbet — Visual Meetings

· IDEO U — Storytelling for influence

· Scrum Manager BoK — Journey map

BLOCK G · FROM THE USER TO THE SYSTEM

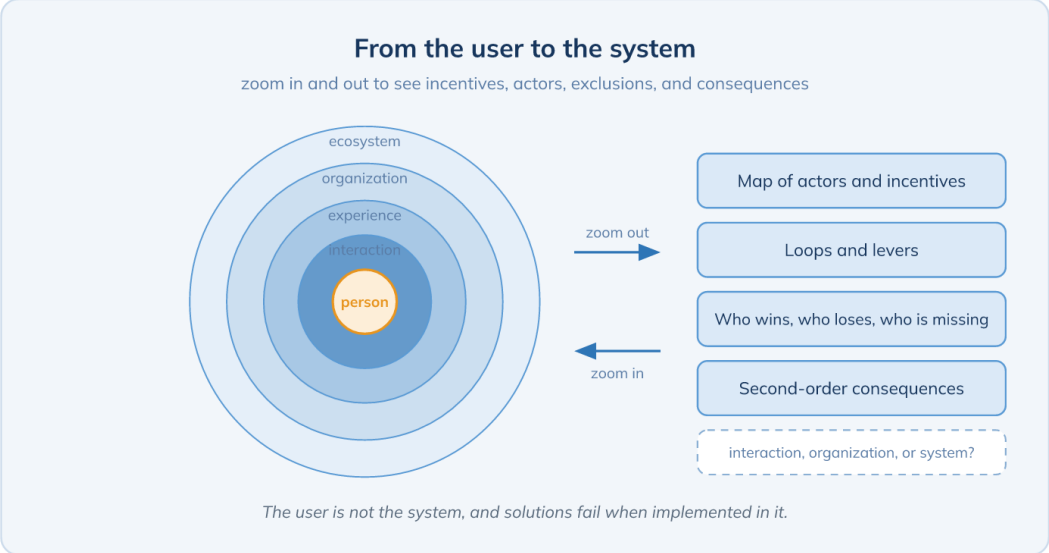
Chapter 21. The limits of the user-centered approach and the actor map

· CONSOLIDATING

Why the user is not the system, where solutions really fail, the extension of human-centered design to humanity and the planet, and the bridging technique all the programs share.

Introduction

Design Thinking was born user-centered and that was its merit against the engineering that designed for itself. But a user is not a system, and solutions that work for one person fail when they collide with the incentives of others, with the structure of the organization, or with what could not be seen from the worktable. This block widens the focus, and does so as the institutions of Design Thinking themselves have done between 2021 and 2025: as an extension of the approach, not as its replacement. In this chapter we see why it is needed and which technique to start with.



21.1 Where solutions fail

Tim Brown and Jocelyn Wyatt wrote it in 2010, in the founding text of Design Thinking for social innovation: a water treatment plant in India produced clean water that families rejected, because the jug could not be carried on the head and the schedule did not fit the husbands' working day; the failure was in the distribution system, not in the product, and their conclusion was that "systemic problems need systemic solutions." Don Norman and Pieter Jan Stappers gave the general diagnosis in 2015 in their article on the problems they called DesignX (healthcare, transport, public policy): those problems do not fail at being understood, they fail at being implemented, when political, economic, cultural, and organizational factors overwhelm everything, and that is why the designer cannot stop at the design phase; they have to take part in implementation, in small, incremental steps. The critique of Natasha Iskander and of Fayard and Fathallah adds the dimension of power: who decides, who is left as user and who as expert, and what agenda hides behind the supposed neutrality of the process.

21.2 From people to humanity and to the planet

Norman, the most cited voice of human-centered design, proposed in 2022 and 2023 moving from the individual human being to humanity. His five principles of humanity-centered design are to solve the root causes and not the symptoms, to attend to the entire ecosystem of people, living beings, and environment, to take a systemic, long-term point of view because the harms reveal themselves decades later, to test and refine with the community, and to design with the community and, if possible, to support the designs made by it. The Design Council arrived at the same place from its Systemic Design Framework, with the principle of being people and planet centered. The correct reading for this skill is the one both make: human-centered design is not abandoned, it is extended. The third dimension of inclusive design formulated by OCAD's research center brings it all together in one sentence: whoever designs does so within a complex adaptive system.

Extension, not replacement. Looking at the system does not replace looking at the person. It adds to the question "what does this person need?" three others: who else is affected, what incentives does each actor have, and what will happen when the solution is implemented in their world.

21.3 Zooming in and out, and the actor map

"Zooming in and out" is one of the six principles of the Systemic Design Framework and a practice of service design, which constantly changes zoom level between the end-to-end experience and the detail of a moment. As a framing tool, at Scrum Manager we use a ladder of five levels: the person, the interaction, the experience, the organization, and the ecosystem. A problem can be read at any of them, and choosing the level is a decision, the same one chapter 6 described about symptoms and causes.

The technique to start with is the only systemic one that all Design Thinking programs share: the **actor map** or stakeholder map. LUMA, IBM, and IDEO's kit have it, and the systemic design toolkit puts it at the start of its method. A useful actor map is not a list of departments: it records who wins and who loses with each solution, what incentives each actor has, who can veto the implementation, and who is not on the map because nobody thought of them. The Scrum Manager BoK records a brief facilitation technique, the perspective pause, for a team to adopt for a few minutes the point of view of another actor. With the map done, the next block adds the minimum of systems thinking a product team needs.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain with cases why user-centered solutions fail in implementation, in incentives, and in power.
- State the principles of humanity-centered design and present them as an extension of the human-centered approach.
- Zoom in and out between person, interaction, experience, organization, and ecosystem, and choose the appropriate level for a problem.
- Build an actor map with incentives, winners, losers, vetoes, and absentees.

Common mistakes and antipatterns

Designing for the user and forgetting the rest. The water jug worked for whoever designed it and not for whoever had to carry it; the actor map would have shown it.

Stopping at the design phase. The limit of the approach is in implementation; if the team delivers ideas and leaves, the system digests them without changing.

Treating the system as an excuse. "It's a systemic problem" can paralyze; the actor map and small steps are the way to start.

A map of departments. Without incentives, vetoes, and absentees, the actor map is an org chart and says nothing.

Further reading

· Norman — Moving from humans to humanity

· Norman and Stappers — DesignX

· Brown and Wyatt — Design Thinking for social innovation (SSIR)

· Design Council — Systemic Design Framework

· Inclusive Design Research Centre — The three dimensions of inclusive design

· Scrum Manager BoK — Perspective pause

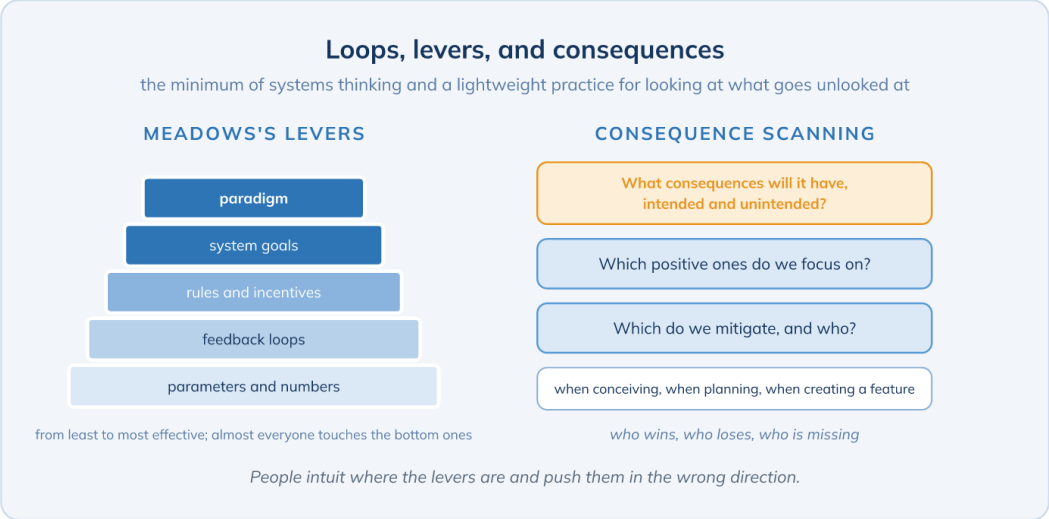
BLOCK G · FROM THE USER TO THE SYSTEM

Chapter 22. Minimal systems thinking, impacts, exclusions, and consequences · CONSOLIDATING

What a product team needs from systems thinking, the responsibility questions with a lightweight practice, deceptive patterns and their regulation, inclusion as a lens, and sustainability as long-term costs.

Introduction

No Design Thinking program teaches systems thinking as a discipline, with causal diagrams and simulation, and neither does this skill. What a product team needs is a minimum: vocabulary for talking about loops and levers, two or three patterns it will recognize in its own backlog, and a lightweight practice for asking who wins, who loses, who is missing, and what consequences it has not looked at. In this chapter we gather that minimum with its tools, all open and mature, and with the legal context that has turned part of design ethics into an obligation. At each point we distinguish what is consensus, what is a trend among the institutions, and what is frontier.



22.1 The minimum of systems thinking

The vocabulary is short: actors, relationships, flows, dependencies, and **feedback loops**, the reinforcing and the balancing ones. On top of it, Donella Meadows left in 1999 the list of twelve leverage points for intervening in a system, ordered from least to most effective, from parameters and numbers (what almost everyone touches) to the system's rules, its goals, and the paradigm they arise from; and she recalled Jay Forrester's observation that people intuit where the levers are and push them in the wrong direction. Peter Senge's archetypes give a name to what a Product Owner will recognize in their product: **fixes that fail**, the patch that solves the symptom today and worsens the problem tomorrow; **shifting the burden**, the symptomatic solution that atrophies the capacity to solve the cause; and the **tragedy of the commons**, the shared resource that each actor rationally depletes. The systemic design toolkit by Namahn and shiftN offers a seven-step method that starts by setting the boundaries of the system and listening to those who live in it. This minimum is a trend among the institutions; formal systems thinking stays out.

22.2 Impacts and consequences: a lightweight practice

The four questions of responsibility in design are who benefits, who may be harmed, who is left out, and what unintended consequences, of second and third order, have not been looked at. The d.school has built them into its teaching with values cards and commitment maps. For an agile team there is a practice tailored to it: **consequence scanning**, which Doteveryone published in 2019 as an "agile event" that fits into the iterative cycle, with three questions (what are the intended and unintended consequences of this feature, which positive ones do we want to focus on, which do we want to mitigate) and three moments (when conceiving the product, when planning the roadmap, when creating a feature); it has an open license. Microsoft's harms modeling is its structured version, with four families of harm (injury, denial of services, violation of rights, including environmental impact, and erosion of social structures) and four assessment factors (severity, scale, probability, frequency). Artefact's tarot cards of tech do the same in game format. These tools exist and are mature; in Design Thinking courses they come in as a mindset, not as a block.

A practice with a name. Asking about consequences without a method produces good intentions. A thirty-minute consequence scan when planning each relevant feature produces a list of mitigations with an owner.

22.3 Deceptive patterns, inclusion, and sustainability

Three matters complete the chapter, with different degrees of adoption. **Deceptive patterns**, which Harry Brignull has cataloged since 2010 in eighteen types, from false scarcity to the cancellation that cannot be found, have stopped being an ethical matter and become a legal one: article 25 of the European Digital Services Act, in force since 2022, prohibits platforms from designing interfaces that deceive or manipulate, and the United States Federal Trade Commission published its report on them the same year. A European Product Owner already designs under that rule.

Inclusion is consensus in user experience training and a lens in Design Thinking training, as this skill treats it. Its minimum vocabulary comes from Microsoft's inclusive design kit: the **mismatch** between people and the experience as the origin of exclusion, the spectrum of people (permanent, temporary, or situational, such as an amputated arm, a broken wrist, or a baby in arms), and the principle of solving for one and extending to many. OCAD's research center distinguishes inclusive, accessible, and universal. And the legal context is recent: the web accessibility guidelines 2.2 have been a recommendation since October 2023 and the European accessibility act has applied since June 2025 to consumer banking, e-commerce, and e-books, among others. Accessibility as a technical specialty belongs to UX training; here the question is who is left out of what we design.

Sustainability, by contrast, is a strong institutional trend and a weak training presence: the Design Council has turned it into a mission and a competence framework, IDEO and the Ellen MacArthur Foundation published a circular design guide, and the W3C maintains web sustainability guidelines. For a product team, what can be defended is treating it as what it is in design: impacts that exceed the immediate experience, long-term effects, and indirect costs that the actor map and the consequence scan must include. The frontier currents, life-centered or more-than-human design, are named in the next chapter.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Use the minimum vocabulary of systems thinking (loops, levers, archetypes) to read a product problem.
- Facilitate a consequence scan when conceiving a product, planning the roadmap, or creating a feature, and record mitigations with an owner.
- Recognize deceptive patterns and explain why in Europe they are already a legal matter.
- Apply the inclusion lens (mismatch, spectrum of people, solve for one and extend to many) and place sustainability as indirect and long-term costs.

Common mistakes and antipatterns

Teaching system dynamics. Causal diagrams and simulations exceed what design work needs; two well-understood archetypes yield more.

Asking about consequences without a method. Without a practice with a name and a moment, the question produces intentions and not mitigations.

Treating deceptive patterns as a matter of taste. They have been illegal in the European Union since 2022 for platforms; designing a hard-to-find cancellation is no longer a business decision.

Confusing the inclusion lens with an accessibility course. Technical accessibility is a UX specialty; design's question is who is left out and why.

Further reading

· Meadows — Leverage points: places to intervene in a system

· Systemic Design Toolkit — Methodology

· Doteveryone — Consequence scanning

· Microsoft — Harms modeling

· Brignull — Deceptive patterns

· Microsoft — Inclusive design

· W3C — What's new in WCAG 2.2

· Directive (EU) 2019/882 — European Accessibility Act

BLOCK G · FROM THE USER TO THE SYSTEM

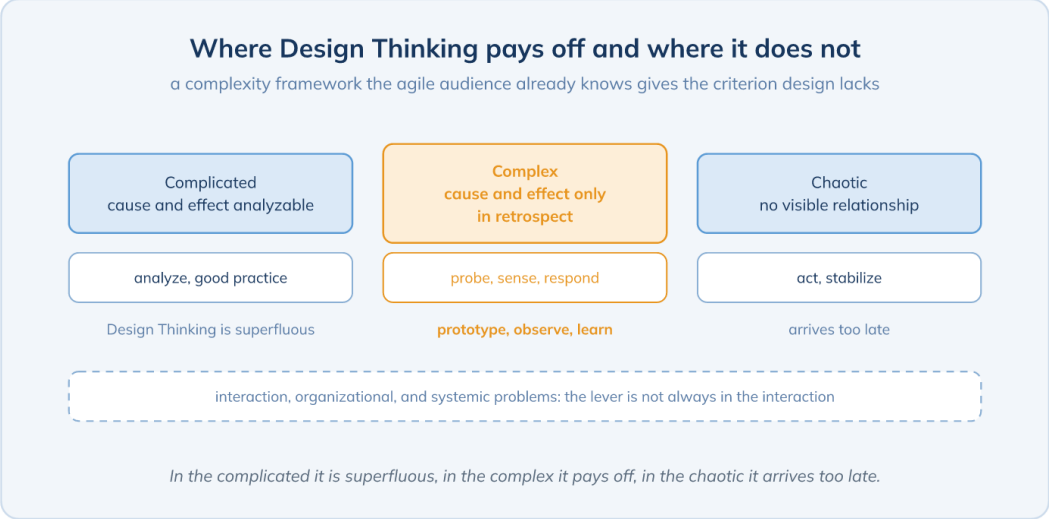
Chapter 23. When to widen the focus and recognize the limits ·

CONSOLIDATING

Interaction, organizational, and systemic problems; a complexity framework for deciding where Design Thinking pays off, is superfluous, or arrives too late; the limits its institutions have declared, and the frontier.

Introduction

Not every problem calls for Design Thinking, and the field itself offers no criterion for knowing it. This chapter imports one that the agile audience already knows, distinguishes three types of problem, gathers the limits that IDEO, the d.school, and the Design Council have acknowledged in writing, and places the frontier currents toward which the conversation is moving, so that the reader knows what is being talked about without taking them for canon.



23.1 Three types of problem

A rule of thumb distinguishes **interaction** problems, in which the lever lies in how a person uses something and Design Thinking pays off with all its practices; **organizational** problems, in which the lever lies in incentives, structure, and decisions of the organization itself, and design work helps to understand and to prototype changes but does not implement them on its own; and **systemic** problems, with many actors, a long time horizon, and nobody in control of the whole, in which design is one voice among several and the reference framework is the Design Council's systemic design. Recognizing the type before calling a workshop saves a lot of theater.

23.2 A framework for deciding

The operational criterion that Design Thinking lacks is provided by Cynefin, the framework by David Snowden and Mary Boone published in 2007, which the Scrum Manager audience knows from its core training. In a **complicated** context, where the relationship between cause and effect exists and an expert can analyze it, Design Thinking is superfluous: analysis and good practice are what is needed. In a **complex** context, where cause and effect can only be seen in retrospect, the response is probe, sense, and respond, and that is exactly prototyping, observing, and learning; there the

approach pays off. In a **chaotic** context it arrives too late, because one has to act first and stabilize. And most of the product problems a team believes to be complicated are complex, which is the reason the approach has so much to offer and the reason it fails when applied to what was complicated. Richard Buchanan had described the same thing from design in 1992 when distinguishing "tame" problems from wicked ones.

Probe, sense, respond. Design Thinking is the practice of the complex domain: in the complicated it is superfluous, in the chaotic it arrives too late, and when the problem is one of organizational structure or public policy the lever is not in the interaction.

23.3 The declared limits and the frontier

The institutions have declared their limits. IDEO admitted in 2010 that systemic problems need systemic solutions and, in 2023, that expertise lies more in the hands of those who live in the system than in the designer's; its commercial answer was to create separate courses on systems thinking and organizational change. The d.school reformed its curriculum toward second- and third-order consequences and ethics from the start. The Design Council admitted with the Systemic Design Framework that the Double Diamond was not enough for the big challenges. Rebecca Ackermann documented the mismatch between the short cycles of commercial design and the long ones of public policy, which explains a good part of the famous failures. The complementarity with strategy, objectives, and organizational change is reasonable (the objectives set the outcome to move, design explores how), although it has no source that formulates it that way.

The frontier, lastly. Design justice, with its 2018 principles, proposes that the affected communities lead. Transition design, since 2015, frames wicked problems in horizons of decades. More-than-human design, which the design research community declared in 2024 a turn of the field, includes non-humans in the process; life-centered and planet-centered design are its applied versions. They are proposals with their own community and no presence in generalist training, and a professional should know that they exist and where they point without taking them for assessable content.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Classify a problem as one of interaction, organizational, or systemic before deciding which practice is appropriate.
- Apply Cynefin to decide whether a problem calls for analysis, Design Thinking, or immediate action.
- Explain the limits that IDEO, the d.school, and the Design Council have acknowledged and what they have done about them.
- Place design justice, transition design, and more-than-human design as frontier, distinguishing them from the canon.

Common mistakes and antipatterns

Applying Design Thinking to the complicated. If an expert can analyze the cause, an ideation workshop is expensive theater; analysis is what is needed.

Promising systemic solutions from a workshop. A problem with many actors and a long horizon is not solved in a week; design is one voice and needs the others.

Taking the frontier for canon. More-than-human design or the pluriverse are proposals with their own community, not what a product professional must know how to do today.

Not saying that it arrives too late. In a chaotic context one stabilizes first; acknowledging it is part of the criterion.

Further reading

· Snowden and Boone — A leader's framework for decision making (Cynefin)

· Norman and Stappers — DesignX

· Ackermann — Design thinking was supposed to fix the world

· IDEO U — Human-centered systems thinking

· Design Council — Systemic Design Framework

BLOCK H · DESIGNING YOUR OWN DESIGN PROCESS, ALSO WITH ARTIFICIAL INTELLIGENCE

Chapter 24. There is no universal sequence: choosing and combining practices · ESTABLISHED

Methods versus process, the precedents of recipes and of the loop, starting where the uncertainty is, the seven questions for choosing a practice, and the criterion for closing an activity.

Introduction

Everything before converges here. A professional who knows how to understand, reframe, diverge, prototype, and co-create still has to decide, in the face of each challenge, what to do first, what to combine, and when to stop. That decision is what the skill calls designing your own design process, and this chapter makes it operational: why there is no universal sequence, what precedents the idea has, where to start, and with which questions a practice is chosen.

24.1 Methods versus process

Chris Pacione, founder of LUMA, put it in one sentence: design is not a process, it is a discipline that underpins your process. Methods are elementary and combine in thousands of ways, and LUMA teaches how to do it with recipes, combinations of two or three methods for a specific challenge (identifying and agreeing on the right problem; generating many ideas and choosing the best; making tangible early to get feedback). The precedents are several. Luke Hohmann published in 2006 a table for choosing the innovation game according to the objective. IBM's loop is explicitly non-linear: teams iterate between observing, reflecting, and making "as needed." And the d.school's Bootleg presents itself as a deck that "can be started anywhere." None of the reference institutions teaches a mandatory sequence today; the courses that copy them do.

24.2 Starting where the uncertainty is

The entry point is not always the same. A team that does not understand the problem starts by observing and synthesizing; one that understands it and has an idea starts by prototyping it in order to learn; one with many ideas and no decision starts by converging; one that does not know whether the problem is one of interaction or of organization starts with the actor map. Bill Buxton gave the clearest version of this criterion for software: one invests in sketches when the uncertainty is what to design (*getting the right design*) and in engineering when it is how to do it well (*getting the design right*). This skill's rule, which has no source that formulates it that way and which we present as a Scrum Manager criterion, is to start where the greatest uncertainty is, and to skip what does not reduce it.

There is no first step. The question "where do we start?" has a different answer for each challenge. The five-step recipe always gave the same one, and that was its problem.

24.3 The seven questions and the stopping criterion

To choose a practice, the team asks what it needs now: to understand (observation in context, interviews, extreme users), to observe more precisely (what, how, and why; shadowing), to synthesize (download, affinity, insights), to imagine (hybrid mode with its levers), to decide (criteria,

matrices, decider), to make tangible (sketch, prototype of just-enough fidelity), or to learn (test with people, record, and decision). Each question points to a block of this guide and a few practices within it. And they are combined according to the problem and not according to the method: one session may need only synthesis and reframing; another, only prototype and test.

The stopping criterion is missing, and here the literature is silent. There is no source that says when to close an activity, so we formulate it as an ability: an activity is closed when it has stopped reducing the uncertainty it was called for, and one changes practice when another, greater uncertainty appears. Marking in advance what can be dropped when designing the sequence, as SessionLab recommends, and going back, going deeper, or skipping when the challenge asks for it, is iterating one's own process, not failing at it.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain why design is not a process but a discipline that underpins each team's process, with its precedents.
- Choose the entry point of a piece of design work according to where the greatest uncertainty lies.
- Apply the seven questions (understand, observe, synthesize, imagine, decide, make tangible, learn) to choose and combine practices.
- Decide when to close an activity and when to change practice, and mark what can be dropped when designing a sequence.

Common mistakes and antipatterns

Always starting with empathize. If the team already understands the problem, observing again is theater; one starts where the uncertainty is.

Applying the full recipe. Five phases for a challenge that only needed prototyping and testing is the most common way to exhaust a team.

Never stopping. An activity that no longer reduces uncertainty is closed; carrying on out of loyalty to the plan is the opposite of designing the process.

Presenting one's own criterion as canon. "Start where the uncertainty is" and "close when it stops contributing" are Scrum Manager criteria, and are stated as such.

Further reading

· LUMA — Recipes: the power of combining design methods

· LUMA — the 36 methods

· Hohmann — Choosing Innovation Games to meet your goals

· IBM — Enterprise Design Thinking (the Loop)

· d.school — Design Thinking Bootleg

· Buxton — Sketching User Experiences

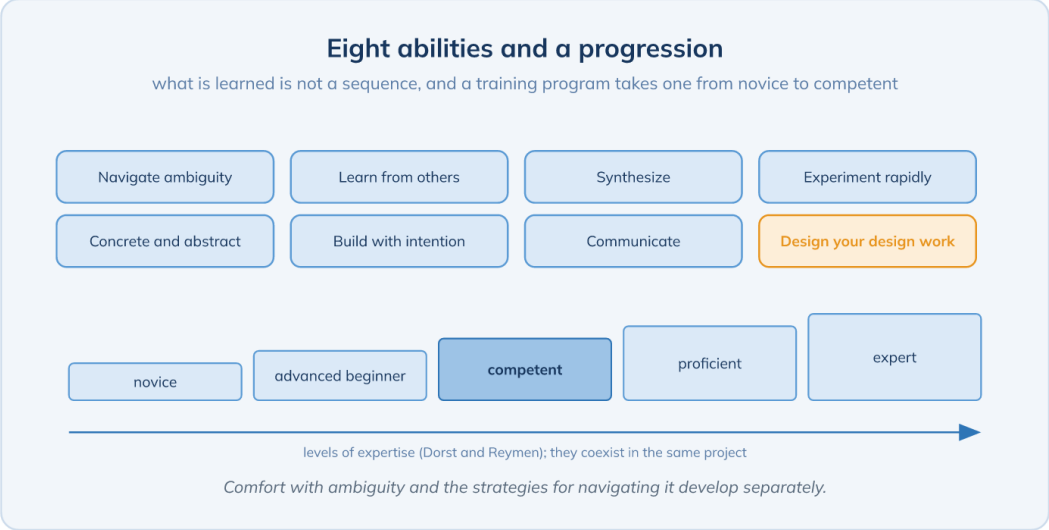
BLOCK H · DESIGNING YOUR OWN DESIGN PROCESS, ALSO WITH ARTIFICIAL INTELLIGENCE

Chapter 25. Design abilities and professional maturity · CONSOLIDATING

The d.school's eight abilities, navigating ambiguity as the cross-cutting ability, the levels of design expertise, and the reasonable progression from novice to competent.

Introduction

If Design Thinking is not a process, what is learned are abilities, and it is worth naming them. The list this skill uses is not its own: it is the one the d.school published in 2016 when it abandoned the hexagon, and it remains active in 2026 with research behind it. This chapter presents it with its source, explains the ability that cuts across the others, and describes, with the literature on design expertise, what progression someone who completes this training can reasonably expect.



25.1 The d.school's eight abilities

Carissa Carter listed in 2016 the eight design abilities the school teaches instead of the five phases. **Navigate ambiguity**: recognizing and persisting in the discomfort of not knowing, holding possibilities in parallel, and resolving or emerging from ambiguity when needed. **Learn from others (people and contexts)**. **Synthesize information**. **Experiment rapidly**. **Move between concrete and abstract**. **Build and craft intentionally**. **Communicate deliberately**. And **design your design work**, which the school defines as recognizing a project as a design problem and choosing the appropriate people, tools, techniques, and processes to tackle it. Each block of this guide exercises several; the previous chapter is the eighth.

Navigating ambiguity is, in the words of the school and of David Kelley, the most important ability and the one that cuts across the other seven. In March 2026 the d.school published an instrument to measure it and a finding useful to anyone who trains: comfort with ambiguity and the strategies for navigating it develop separately; one can be comfortable without having strategies and have strategies without being comfortable. A skill teaches strategies, and should not promise comfort.

25.2 The levels of design expertise

Kees Dorst and Isabelle Reymen applied in 2004 the Dreyfus brothers' model of skill acquisition to design, with seven levels. The **novice** considers the features of the situation as given and follows strict rules. The **advanced beginner** is sensitive to exceptions and uses maxims. The **competent** practitioner selects the relevant elements of the situation, chooses a plan to achieve their goals, is much more involved, works by trial and error, and feels a clear need to learn and reflect. The **proficient** practitioner immediately sees what matters. The **expert** responds intuitively, and many professionals never go beyond that. The **master** sees the standard ways of working as contingent, and the **visionary** extends the domain. Two nuances from the authors matter here: the levels coexist within the same project, because the same person follows rules in some parts of their work and interprets in others; and the model does not describe everything that is learned, because design also requires declarative knowledge. Nigel Cross added a warning for those who teach: what designers know about their own processes is largely tacit, and the teacher has the responsibility to be as explicit as they can.

From novice to competent. A training program reasonably takes one from the novice who follows the recipe to the competent practitioner who chooses a plan, gets involved, and reflects. The expert who responds intuitively is made only by practice, and promising otherwise would be untrue.

25.3 The progression of this skill

With that framework, the professional maturity this skill pursues is described in three movements that correspond to the step from novice to competent: from executing techniques to selecting them according to the problem; from following a process to designing it; and from producing artifacts (maps, notes, prototypes) to generating learning that changes decisions. The outward signs are recognizable: the competent practitioner can explain why they chose one practice and not another, can say when an activity has stopped contributing, and leaves a session with findings and decisions and not just with a full wall. The abilities framework has been official at the d.school since 2016 and the research on expertise has backed it since 2004, but mainstream training still certifies the execution of the phases; that is why this chapter carries the label it carries.

What you should be able to do

Having mastered this chapter, a professional is able to:

- State the d.school's eight design abilities with their source and recognize which one each practice in the guide exercises.
- Explain why navigating ambiguity cuts across the others and why a training program teaches strategies and not comfort.
- Place one's own work in Dorst and Reymen's levels of expertise and recognize that they coexist in the same project.
- Describe the progression from novice to competent in three movements and recognize its outward signs.

Common mistakes and antipatterns

Presenting the abilities as one's own framework. They are the d.school's since 2016; citing them with their source is honest and strengthens the thesis.

Promising the expert level. Intuition without distinguishable reasoning is given only by practice; a skill takes one to competent.

Confusing comfort with ability. Being comfortable with ambiguity and knowing how to navigate it are different things; the second is what is taught.

Measuring maturity by artifacts. A wall full of notes is not maturity; leaving with findings that change decisions is.

Further reading

· Carter — Let's stop talking about THE design process (d.school)

· d.school — Navigating ambiguity, the super ability

· d.school — measuring the ability to navigate ambiguity (2026)

· Dorst and Reymen — Levels of expertise in design education

· Cross — Designerly ways of knowing

BLOCK H · DESIGNING YOUR OWN DESIGN PROCESS, ALSO WITH ARTIFICIAL INTELLIGENCE

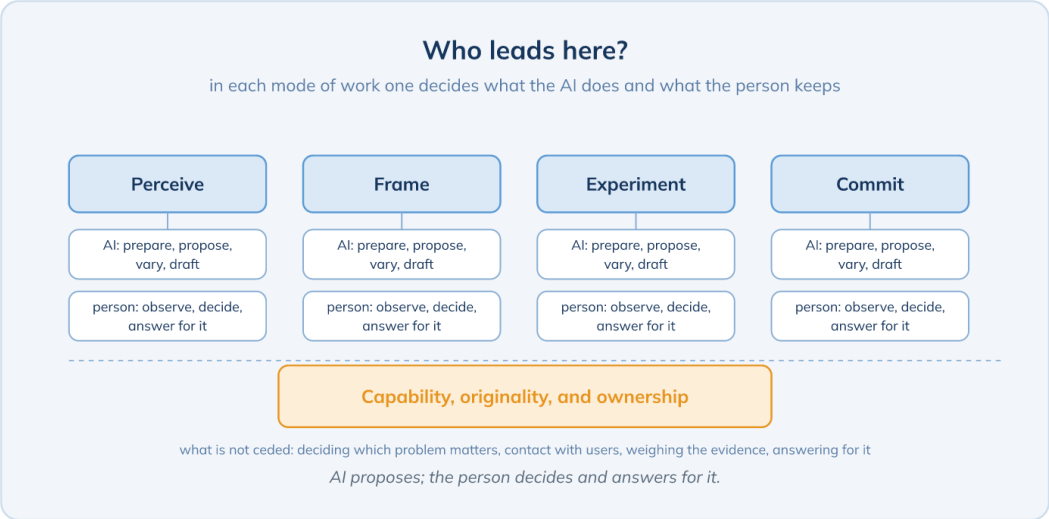
Chapter 26. Artificial intelligence in each mode: who leads here? ·

EMERGING

Using AI as a process decision, the question of who leads at each moment and in three dimensions, what AI does well in design, and what is not delegated.

Introduction

The Product Owner 2.0 guide and the AI applied to agile work skill already teach the general principle for using artificial intelligence, verification, confidentiality, and the criterion of what is delegated and what is not. This skill does not repeat it. What it adds is the question specific to design work: if designing your own process is the central ability, deciding what the AI does in each mode of that process is the newest process decision, and it has to be made consciously so that nobody else makes it. In this chapter we present the most useful framework we have found for doing so, what AI does well in each mode, and what is not delegated.



26.1 Agency drift

The HPI School of Design Thinking in Potsdam published in August 2026 a framework it calls co-agency design, which starts from a phenomenon with a name: **agency drift**, the gradual migration of influence and decision power toward artificial intelligence without anyone having decided it. It happens when the team asks the AI for ideas before generating its own, accepts its clustering of notes without discussing it, or takes its prototype as the starting point and not as an alternative. The framework's answer is a question asked at each moment of the process, "who should lead here?", which is answered differently when perceiving, framing, experimenting, and committing, in three dimensions that must be preserved: human **capability**, so that skills do not atrophy; **originality**, so that the AI widens possibilities instead of homogenizing them; and **ownership**, so that it is clear who answers for the decision. It maps almost one to one onto the core idea of this skill, understand, reframe, make tangible, learn, and turns the principle of use into a process design decision rather than a list of prohibitions.

26.2 What AI does well in each mode

IDEO U has taught since 2025 three courses on design with AI, with explicit caveats, and its framework for AI in research sums up the legitimate uses: broadening perspective, perceiving nuance, challenging assumptions as a dialectical partner that introduces friction, and making ideas concrete quickly, on condition that meaning-making remains human. Nielsen Norman Group, with its own studies, details what it speeds up: in research, planning, transcription, translation, preliminary coding, and writing, and not observing tests or moderating studies; in workshops, setting the evaluation criteria before generating, devoting eight or ten minutes to assisted ideation, and co-writing the prompts in groups of two to four people so as not to lose the conversation. The only experimental evidence on AI as facilitator, a study with 1,475 participants published in 2026, shows that it raises the participation of those who speak least and does not improve the quality of the decision.

By mode, what AI does well is concrete. When understanding, coding with given criteria and proposing a first clustering (chapter 8). When reframing, generating variants of a question and alternative frames that the team judges. When diverging, expanding variations and combinations after the human round, with prompts that force diversity (chapter 11). When making tangible, prototyping fast and in several directions, degrading fidelity on purpose when appropriate (chapter 16). When co-creating and facilitating, preparing the agenda and materials, proposing, and documenting (chapter 18). In none of them does it decide.

AI proposes; the person decides and answers for it. Scrum Manager's general principle, applied to design, becomes a question per mode: at this moment, who leads, and what capability, originality, and ownership do we preserve when answering it?

26.3 What is not delegated

The list of what is not delegated is short and stable, and coincides with that of the Product Owner 2.0 guide (what decides, relates, or answers for something is not delegated; the AI does not sign), to which we refer for the general principle. In design work it takes this form: deciding which problem matters and which challenge is explored; contact with users and, all the more so, with people of identities different from the team's, because that is where the models fail most; weighing the evidence and deciding what is a finding; interpreting the context; closing disagreement, which is where a session's learning lies; and answering for the decision. Everything else can be delegated with review. Verifying what the AI produces and the confidentiality of users' data are covered in the AI skill and in Product Owner training.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain agency drift and ask the question "who leads here?" in each mode of work, preserving capability, originality, and ownership.
- Assign to AI what it does well in each mode (coding and proposing, varying and combining after the human round, prototyping in several directions, preparing and documenting).
- List what is not delegated in design work and explain why contact with people of different identities is the most sensitive case.

- Place the general principle for using AI in Product Owner training and in the AI skill, and apply here only what is specific to design.

Common mistakes and antipatterns

Letting the AI lead without deciding it. Agency drift is gradual and invisible; the question of who leads is asked at each moment, not once.

Using AI as a list per phase. "AI for empathizing, AI for ideating…" turns a process decision into a catalog of tools.

Taking its critique for a decision. The AI critiques fluently, but there is no evidence that its critique moves the group; its best use is giving an anonymous voice to dissent.

Repeating the general principle. Verification, confidentiality, and prompting are already in other Scrum Manager training; here what is specific to design is applied.

Further reading

· HPI School of Design Thinking — Co-Agency Design

· IDEO U — IDEO's framework for using AI in research

· NN/g — Accelerating research with AI

· NN/g — Facilitating AI-enhanced workshops

· Alsobay et al. — AI facilitation of groups (CSCW 2026)

· Skill Arena — Product Owner 2.0

· Skill Arena — AI applied to agile work

BLOCK H · DESIGNING YOUR OWN DESIGN PROCESS, ALSO WITH ARTIFICIAL INTELLIGENCE

Chapter 27. Risks specific to artificial intelligence in design · EMERGING

Homogenization, flattening of identities, manufactured evidence, instant fidelity, skill loss, intellectual property, and the transparency duty, distinguishing what has evidence and what is an indication.

Introduction

The general risks of artificial intelligence, hallucinations, automation bias, privacy, are already covered by Product Owner training and the AI skill. Those in this chapter are the ones that take a form of their own in design work, and most have recent and convergent evidence. We present them with their source and with a distinction the skill itself demands: what is proven, what is an indication, and what is still a reasonable inference.

27.1 Homogenization: the best documented risk

It is the risk most specific to design because it affects the core of divergence, and the best documented. Anil Doshi and Oliver Hauser showed in *Science Advances* in 2024, with 293 writers and 600 evaluators, that access to AI-generated ideas improves individual creativity, especially that of the less creative people, and at the same time makes the stories more alike: a social dilemma in which each individual improves and the whole grows poorer. Anderson, Shah, and Kreminski found the same in ideation with ChatGPT, and moreover that participants felt less responsible for their ideas. Meincke, Mollick, and Terwiesch measured that GPT-4's sets of ideas are less diverse than those of human groups and that step-by-step reasoning recovers part of the diversity; the study by Terwiesch and colleagues published in 2026 confirms that AI ideas are rated better and are more alike. Lee and Chung added that AI performs better on incremental ideas than on radical ones. The remedy follows from the evidence and no source prescribes it as a norm: the human round first, the AI afterwards and to vary, with prompts that force diversity, and measuring categories, not ideas.

27.2 Synthetic personas and manufactured evidence

Synthetic personas flatten and misrepresent. Angelina Wang and colleagues showed with 3,200 human participants, four models, and sixteen demographic identities that the models portray groups from the outside (*misportrayal*) and homogenize the internal diversity of each group (*flattening*), and that mitigation techniques reduce the problem without eliminating it. Bisbee and colleagues found that synthetic data reproduce the averages of opinion but deviate in almost half of the statistical relationships and in one in three with the sign reversed, with a variance far below the human one and results that change between versions of the model. Agnew and others, at the reference conference on human-computer interaction in 2024, pointed out that replacing participants with AI clashes with the values of representation and inclusion that found participatory research. This is why contact with identities different from the team's is what is least delegated, and why presenting synthetic findings as research is manufacturing evidence, the research theater that the Product Owner 2.0 guide already names.

What AI flattens. A synthetic persona of a minority group is the case where the model fails most and where the harm is greatest. It serves to rehearse questions; never to speak on anyone's behalf.

27.3 Fidelity, skills, ownership, and transparency

Instant fidelity anchors on the first generated solution and makes the surface look final without logic underneath; chapter 16 develops it with the observations of NN/g and Thoughtworks. Skill loss is a risk distinct from the loss of judgment and affects what design does with its hands: observing, synthesizing, sketching. The evidence is consistent in direction and weak in design: Kumar and colleagues showed in 2025, with 1,100 participants in preregistered experiments, that AI assistance improves creative performance while it lasts and harms it when the person goes back to working without it; the studies on critical thinking and cognitive offloading are correlational or small-sample; and the first solid field data on skill loss from exposure to AI (a six-point drop in adenoma detection by endoscopists after months of using AI, published in *The Lancet* in 2025) comes from medicine and not from design, where the effect is plausible and not measured. It should be said that way, without alarm and without denial; the Product Owner 2.0 guide already has the rule of keeping the muscle, one's own analysis first and the AI's afterwards.

Two more matters are already normative. The United States Copyright Office established in January 2025 that prompts to a model are not enough on their own to obtain protection, so that a storyboard or a mood board generated with AI is not registered as one's own creation without substantial human work, and IDEO has recommended since 2023 attributing what is generated and including the prompt alongside the artifact. And article 50 of the European AI Act, applicable since August 2026, requires informing people that they are interacting with a machine, which reaches any conversational prototype tested with users outside the lab. Product Owner training covers the regulation of AI products; here it is enough to know that the prototype is also inside.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain homogenization with its evidence and apply its remedy: human round first, AI afterwards and to vary, measure categories.
- Set out why synthetic personas flatten and misrepresent identities and why presenting their answers as research is manufacturing evidence.
- Distinguish, for each risk, what is proven, what is an indication, and what is inference, and say so to the team.
- Apply the obligations of attributing what is generated and of transparency for conversational prototypes.

Common mistakes and antipatterns

Measuring ideation with AI by quantity. AI produces more and better on average; what the whole loses is variety, and that is measured by counting categories.

Asking the AI about a minority group. It is where it flattens and misrepresents most; real contact with those people is what is least delegated.

Alarming about or denying skill loss. The direction of the evidence is consistent and its design is weak; keeping the muscle is the prudent rule.

Testing an agent without notice. A conversational prototype with people outside the lab must declare that it is a machine since August 2026.

Further reading

· Doshi and Hauser — Generative AI enhances individual creativity but reduces collective diversity

· Anderson, Shah, and Kreminski — Homogenization effects of LLMs on creative ideation

· Kumar et al. — Human creativity in the age of LLMs

· Wang, Morgenstern, and Dickerson — LLMs that replace participants misportray and flatten identity groups

· NN/g — Synthetic users

· European AI Act — article 50

· US Copyright Office — Copyright and Artificial Intelligence, Part 2