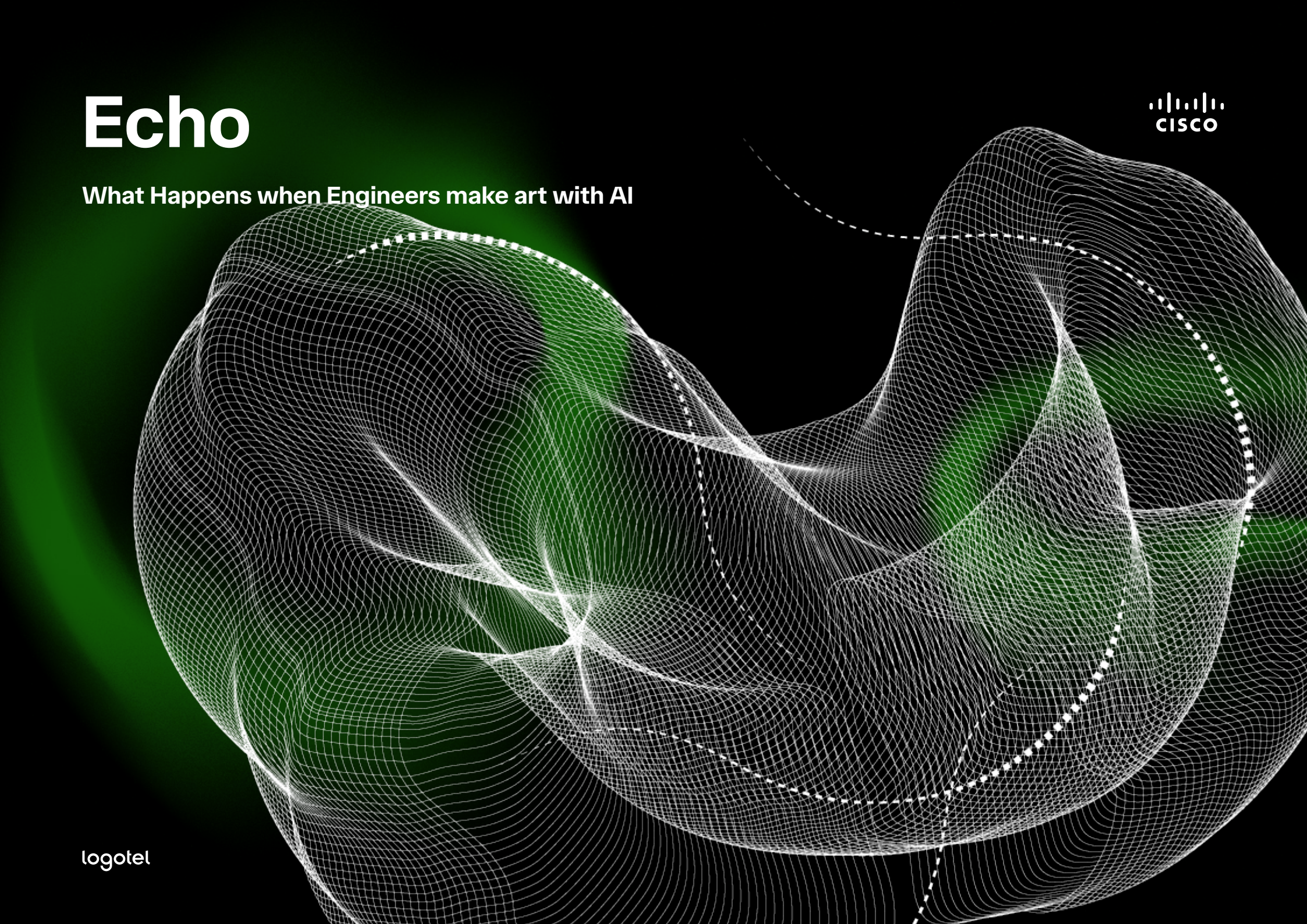
Echo

What Happens when Engineers make art with AI



logotel

Index

FOREWORD / 3

CREDITS / 5

METHODOLOGY / 6

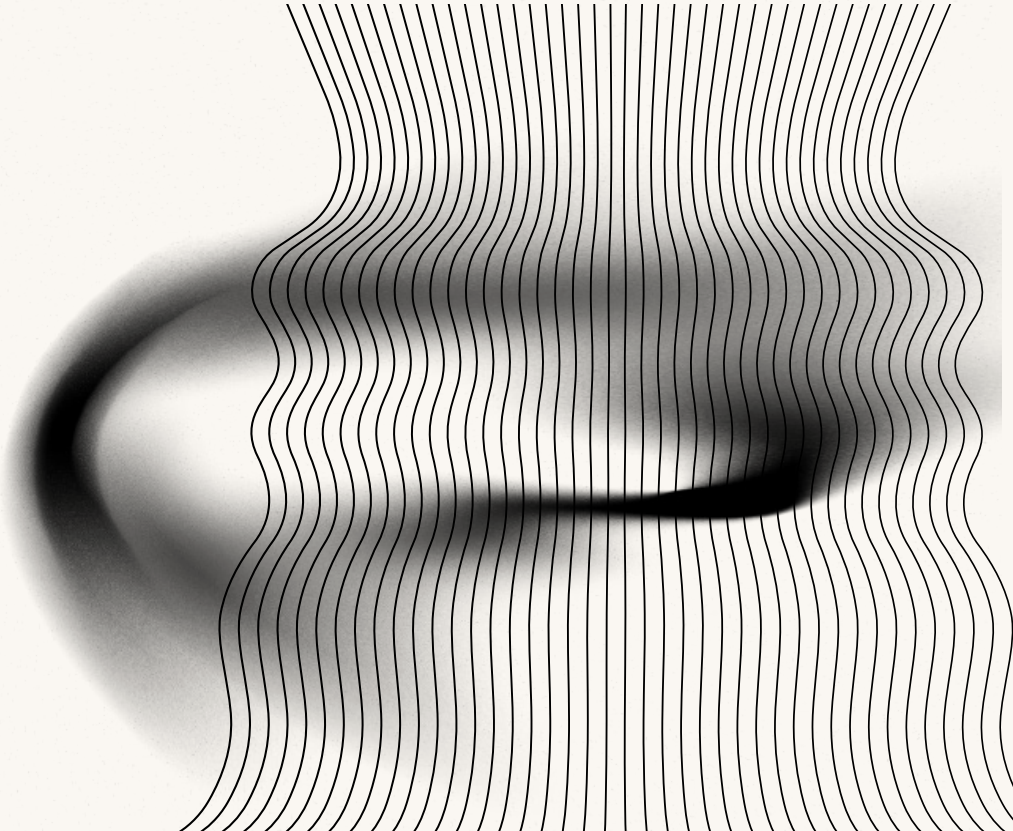
PATTERN 1 · INTENTIONAL PROMPTING / 8

PATTERN 2 · THE AI ARTIFACT AS SHARED LANGUAGE / 14

PATTERN 3 · DESIGNING THE GOVERNANCE LAYER / 20

PATTERN 4 · THE ANCHOR OUTSIDE THE (AGENTIC) LOOP / 26

APPENDIX · DOES DESIGN THINKING NEED AN UPGRADE? / 31



AI changes the room

AI is often described as a technology shift: more computing power, more data centers, more connectivity, more powerful models. That view is true, and it does not tell the whole story. AI is already radically transforming the social fabric of work: how teams form, how people read each other's intent, how they jointly decide, how expertise flows between them, and how organizations understand and serve their customers.

As AI permeates day-to-day work, digital innovation turns into analog transformation: What happens when a group that has worked together before is using AI to create something complex and meaningful under time pressure? What changes when AI is not treated as a tool at the edge, but as a working presence inside the collaboration? How does AI help engineers understanding the needs of artists and vice versa? Can a shared AI artifact become the place where trust, interpretation, and decision-making happen? Echo was an opportunity to explore these questions. It also gave us a rare chance to revisit a long-standing divide between engineering and the humanities. For centuries, the engineering and the arts have developed separate languages; the computer — and now AI built on top of it — offers both worlds a shared medium again.

Conceived by Cisco with partners H-FARM and SHADO for the 61st Venice Art Biennale, Echo brought five Cisco subject matter experts – three AI software engineers and two user experience designers – into a one-week residency with curator Giovanna Zabotti, artist and scenographer Paolo Fantin, and composer Dardust. Their joint goal was to design an immersive installation for the Pavilion of the City of Venice, with AI deeply integrated into both the design journey and the final visitor experience.

By bringing together eight people who had never met before, in a context far removed from their usual professional routines, Echo went beyond an art-and-technology experiment. It became a field lab for exploring how AI changes collaboration between teams, partners, and customers, and how the future of work in an AI-native environment might look and feel. Thereby Echo mirrored a challenge familiar to many customer engagements: technical teams are increasingly asked to respond to needs that are experiential, contextual, and impossible to specify in advance.

To capture what happened inside Echo, Cisco invited a team of ethnographers from Logotel, a service design company, to join the residency as "embedded observers".

Logotel's investigation combined session recordings, Webex exchanges, GitHub activity, interviews, and observation to examine how the group organized itself, how roles formed and

Six days of embedded ethnographic observation surfaced four transferable patterns for AI-native collaboration: how teams form, decide, coordinate, and create – and how organizations can design for those changes

shifted, how AI mediated across disciplines and people, and which new collaborative behaviors emerged.
The results are presented in this document and can be summarized as four transferable patterns of AI-native collaboration:

1. Intentional prompting – effective AI use begins before the prompt. The strongest outputs came from practitioners who first built context: understanding the challenge, the constraints, and the behavior of different AI models. Prompting meant developing the right judgment to ask better questions.

2. The AI artifact – AI made it possible to build and modify a functional artifact within a day. From then on, it became the object the group used as a visible, audible, and changeable object to organize and enrich their collaboration.

3. The governance layer – AI's generative variability had to be actively designed, not just accepted or ignored. The team needed visible controls that translated human intent into bounded AI behavior: balancing freedom to generate with the constraint to stay coherent.

4. The anchor outside the agentic loop – even in highly AI-enabled workflows, the group needed trusted human perspectives grounded in context, taste, and consequence. The generative loop produced endlessly. The anchor helped determine what is right for the situation, the audience, and the intended experience.

Together, these patterns describe something larger than a new way to use AI tools. They point to a new shape of collaboration: faster, more artifact-centered, more cross-disciplinary, and more dependent on intentional human judgment than conventional narratives about automation suggest.
In Echo, AI acted as a translator across two divides. Inside the team, it helped turn artistic, technical, and curatorial languages into visible artifacts that people could react to concretely. Engineers could respond to sensory and poetic direction through code; artists and curators could respond to technical work through what they saw, heard, and felt in the artifact itself. The second divide was between the installation and its audience. In the finished work, AI translated visitors' emotional input into projections, sound, and movement within the immersive space.
The same mediating role that helped the team collaborate also became part of the visitor experience.
The patterns documented in this paper are therefore not only about Echo. They describe dynamics any organization can recognize as AI enters collaborative work: the need to build context before generating, to use artifacts as shared language, to govern AI's variability, and to preserve human judgment outside the momentum of the agentic loop.
For Cisco, Echo is a signal of a broader commitment.
Building infrastructure for the AI era is essential. Understanding what AI does to work, creativity, customer relationships, and human collaboration is just as important.
With Echo Cisco explores that frontier not from a distance, but by entering an unfamiliar environment and studying what happens when technology becomes part of how people make meaning together and build new worlds for others.

Echo shows how AI can become a shared medium where engineering, art, and human experience meet – and where new forms of collaboration can be observed, shaped, and learned from

Credits

Venice Pavilion
Maurizio Carlin – Commissioner
Giovanna Zabotti – Curator

Artists
Paolo Fantin – Set designer
Dardust – Pianist & composer

Cisco – Echo's co-authors
Gianpaolo Barozzi - VP, Technology Officer
Jan Brabec – Principal AI Researcher
Georgina Burton – UI Designer
Kai Cheung – AI Tech Lead & Researcher
Tom Lamberty – Sr Tech Solution Consultant
Edgar Uribe – Software Engineer Architect
Nick Zolfo – AI Integration & Maturity

Shado – Echo's co-authors
Davide Bartolucci – CEO
Dario Plée - Project Direction
Stefano Bartolucci - Project Manager
Luca Bianco - Full Stack Developer
Serena Vanzillota - Literary Curator
Leonardo Nieto - Animator
Federico Fabiano - Animator

H-FARM – Echo's sandbox
Riccardo Donadon – Founder & CEO

Logotel – Echo's ethnographers
Cristina Favini – General Manager & Chief Design Officer
Valeria Crociata – Sr Visual Designer
Guido Dallago – Visual Designer
Giuseppe Guariniello – Sr Design Researcher
Alice Paracolli – Design Researcher
Vincenzo Scagliarini – Head of Research



Methodology

The research, analysis, and synthesis of this white paper were conducted by Logotel. Three ethnographic researchers participated in all activities alongside the Cisco team during the Echo residency (Venice, 11–16 March 2026) and the post-residency period through May.

Field observation was read against three technical sources: the audio of the sessions recorded through Webex, the team's Webex chat and the project's code repository, hosted on GitHub. Each pattern was identified where these sources converged, so that a behavior noticed in the field could be checked against the record, and a signal in the data could be traced back to what people were doing.

Data sources and analysis. The evidence base combines four types of primary material.

1. Field observation. Logotel's embedded observers documented the residency in real time across six days, producing daily notes (3,500 lines), meeting transcripts, and a structured observation protocol. These were the primary source for identifying behavioral patterns, moments of friction, and decision points that did not surface in technical logs.

2. Audio transcription. Approximately 14 hours of recorded sessions were transcribed using OpenAI Whisper (small model). The resulting 129,639 words were searched for keywords tied to each pattern (prompt, parameter, prototype, color, Paolo [Fantin], GitHub) to locate relevant passages and extract verbatim quotes. Whisper small has known drift on proper names and technical terms; quotes used in the white paper were verified against the original audio.

3. Repository analysis. The project's GitHub repository was queried directly: total commits (504), commits per residency day, AI-assisted commits identified via Co-Authored-By tags (318, 63% of total), and codebase snapshots at discrete dates to trace how the artifact evolved. Commit messages were read to identify moments where external feedback reached the code.

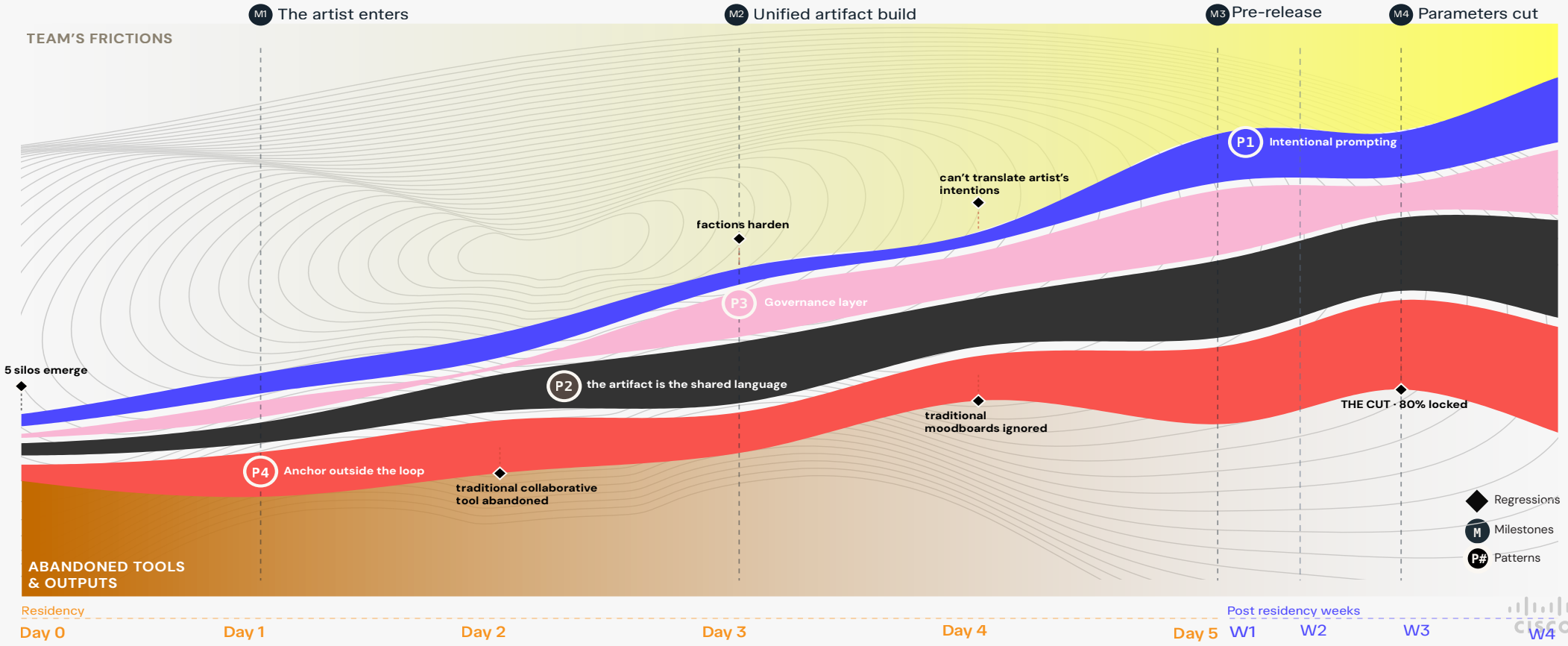
4. Webex chat. 581 messages from 13 participants over 10 weeks were queried via Cisco Webex developer's API.

A note on the numbers. The percentages and counts used throughout this paper, prompts integration ratios, parameter pruning rates, and similar figures, describe what happened in this specific team over this specific project. They illustrate some patterns, not a statistical claim generalizable beyond this case.

Data analysis / 0 How patterns emerged in a novel environment

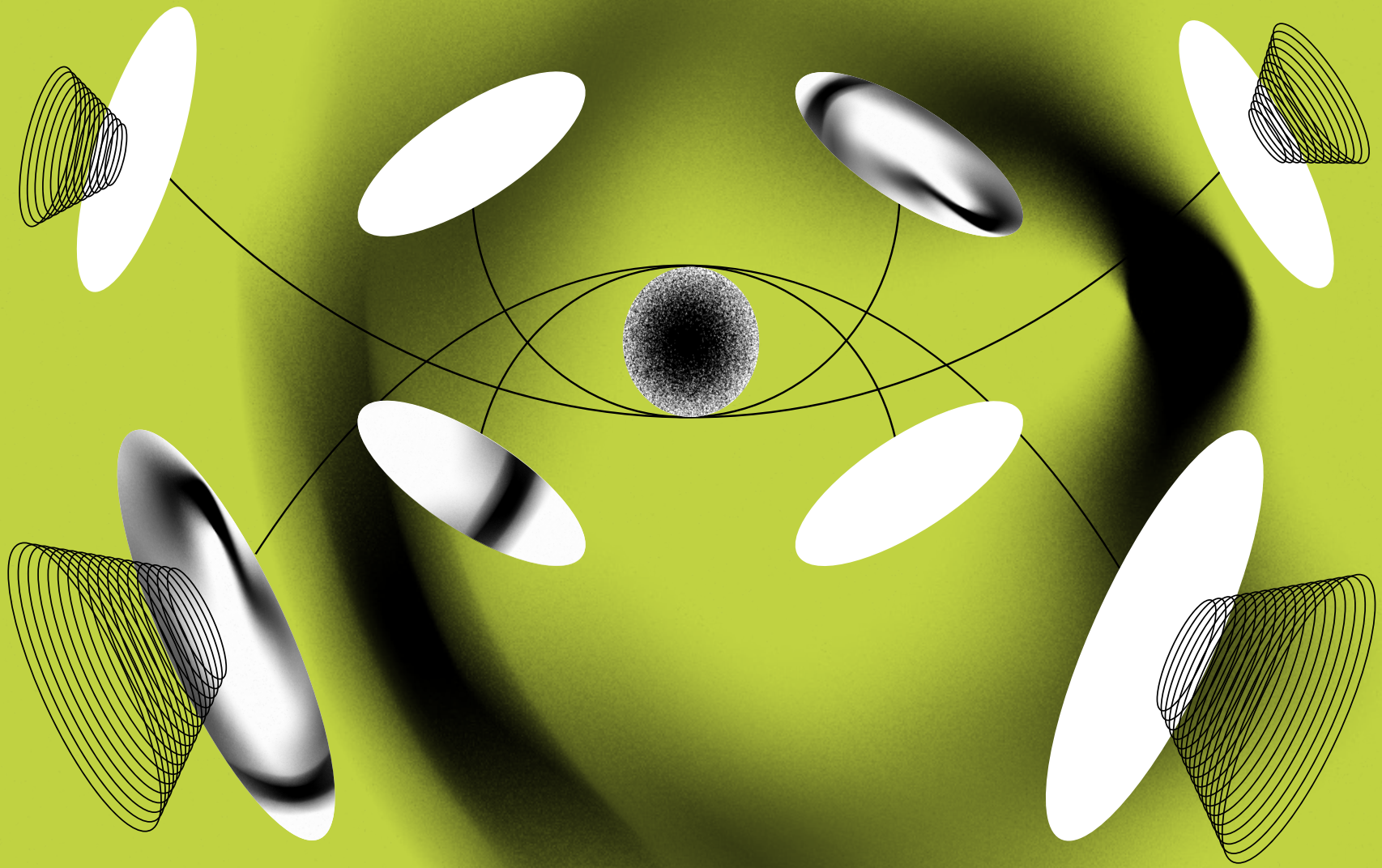
The chart shows how the work consolidated into four patterns over the course of the residency: the X axis is what contributed to the final artifact. It starts narrow (bottom-left) and widens as the patterns take shape. Each band is one pattern; its width is the share of work it accounts for. The yellow space above the stream represents relational frictions: independent work, hardening factions, communication gaps between roles that were never fully resolved. The brown space below represents the abandoned outputs: tools dropped, parameters removed and options left behind.

The patterns did not arrive together. Pattern 4 enters first on Day 1, when the anchor (the artist) reaches into the loop with the first meeting. Pattern 2 crystallizes on Day 3 with the convergence into a single shared artifact. Pattern 3 takes shape on Day 3 with the governance layer. Frictions peak on Days 4 and 5, when the translation problem between the artist's language and the parameters is most acute: "can't translate", "factions harden", "mood board ignored" all appear there. Frictions and abandoned outputs never disappear, but the work reached delivery anyway.



Intentional Prompting

1



WHAT IS IT?

Intentional prompting means researching before generating, taking the time to understand the brief and the project context before asking AI to produce anything.

Intentional prompting begins with two forms of research.

The first is understanding the problem: its constraints, what has already been tried, what would count as a relevant answer.

The second is selecting the AI models best suited to each task and understanding their dynamics: how they generate, why their outputs vary, where their probabilistic behavior is stronger or weaker, and what specific capabilities they offer. Neither investigation is ever complete. Models change frequently, and their behavior can shift across versions, modalities, and use cases. Ongoing exploration of both the problem space and the models helps teams maintain a current and practical understanding of the tools over time.

Together, these two forms of research make prompting intentional. They combine specific knowledge of the problem with an understanding of how models produce results, enabling teams to formulate requests that are relevant, specific, and testable, and to progressively provide the model with the context needed to generate outputs aligned with the project.

Through this double orientation, research exposes the team to domains beyond their own. Over time, it builds the project understanding needed to evaluate AI outputs critically and to engage stakeholders in conversations about the project's direction and outcomes, rather than technical delivery alone.

Intentional prompting

How to implement it

• Interrogate the brief[1]

This involves questioning the assumptions the brief takes for granted, its constraints, and what has already been done. Targeted follow-up questions with the client help identify root causes and clarify the decision to be made. The expected outcome should be defined in terms of end-user experience and perceived value, rather than technical deliverables, so the team has a qualitative target against which to evaluate outputs. Where feasible, quantitative targets, such as speed, quality, and value at risk, should also be identified.

• Create the space to match project needs with model capabilities

Before prompting begins, build an explicit step in which the team breaks down the brief into workstreams and identifies the models best suited to address them.

• Test models dynamics and experiment[2]

Working across models is how people develop the judgment to recognize what works, which models best fit the problem, and which approaches are most likely to improve the quality of the work or open up a new direction. This competence comes from testing, comparing and evaluating multiple AI models and their outputs in real projects.

• Chain the best results together[3]

Teams can query the same concept across systems and assemble a sequence of outputs: a visual reference from one model, a research paper from another, and a code prototype from a third. These components can then be loaded into a structured context folder in the AI tool, alongside briefs, constraints, and decisions. This creates a project-specific foundation that the model can draw on when generating. The context should be kept current as the project evolves.

Strategic implications

• Output quality. What an AI model produces depends on the context it receives. When teams research first and provide project-specific context, the model is less likely to fill gaps with its own defaults.

• Wider contribution. Everyone can contribute, not just specialists. Research before generation builds the shared understanding to evaluate AI output and participate in decisions across disciplines.

• Transferable capability. The ability to investigate both the problem and the models, understanding what to do with what, is not project-specific. On the next project, the problem domain resets, but the practice of exploring what each model can do and matching it to the task stays.

PITFALL MANAGEMENT

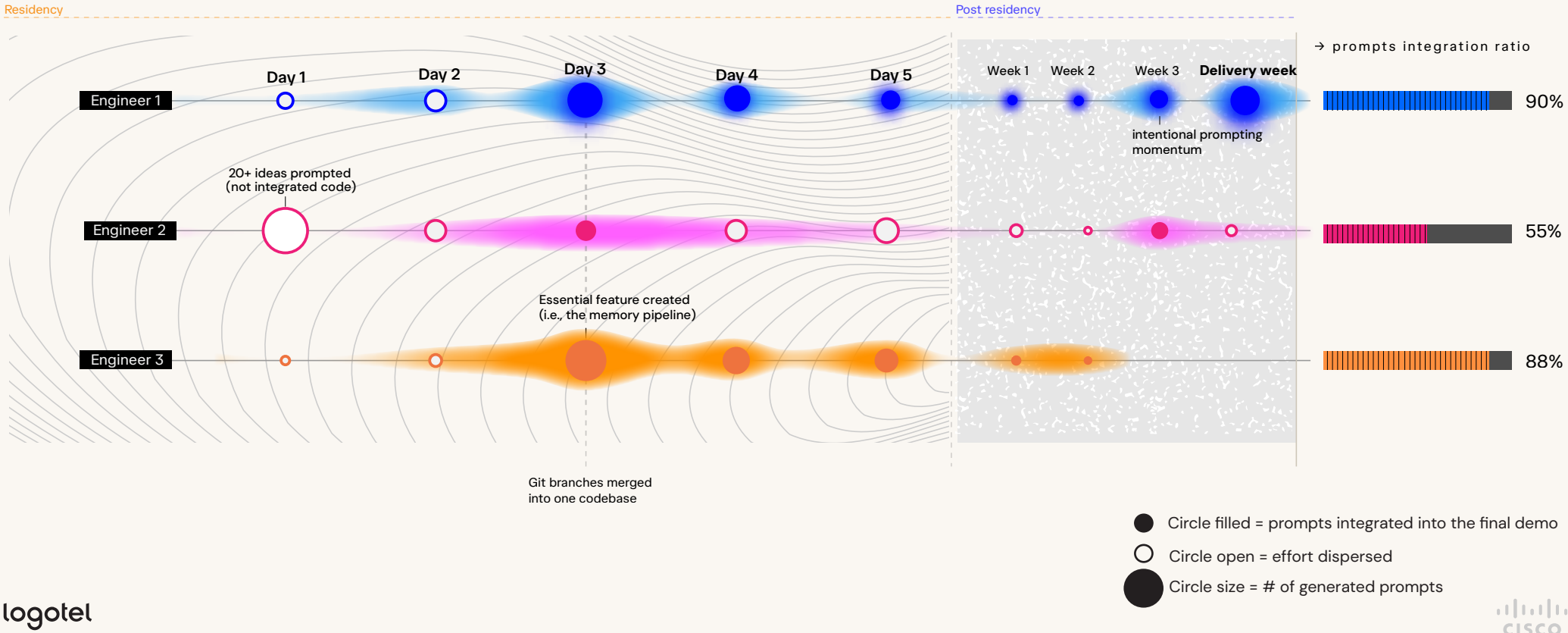
Risk #1. Plausibility without judgment. The value of interrogating the brief lies in establishing a basis for evaluating outputs before generation begins. Without that point of view, there is a risk of plausibility without judgment: the model reflects uncertainty back in a coherent and persuasive form mistaken for insight. The team should be able to describe what a successful output looks like before generation begins, not only recognize it afterwards.

Risk #2. Generation paralysis. As teams compare outputs across multiple models, there is a temptation to keep generating alternatives in search of a better answer. This is the risk of generation paralysis: comparison becomes a substitute for evaluation, and exploration produces more options without increasing understanding.

Risk #3. Context saturation. As teams assemble references, intermediary outputs, research findings, and examples into a shared project context, there is a temptation to keep adding material. Beyond a certain point, additional references stop improving results and begin to introduce ambiguity. This is the risk of context saturation: relevance and coherence matter more than volume. When references conflict, the model tends to mediate between them rather than follow a precise direction.

Data analysis / 1 Intentionality as a proxy to efficiency

The chart visualizes the relationship between intentionality in prompting and its effect on the final demo. The bars on the right measure efficiency directly: how much of each engineer's prompting reached the artifact.
Each row is one engineer's work across the residency; each bubble is a day. Bubble size is prompting depth; filled means that day's work entered the final installation, open means it produced only experiments. Volume and conversion pull apart. Engineer 2 produced the largest burst of the week on Day 2, more than twenty ideas, but not coherent with the project's context, so they stayed outside the installation. Engineers 1 and 3 ran fewer, more directed prompts from the start and a larger share of their work reached the artifact: 90% and 88% respectively, against 55%. Intentional prompting costs time at the start; it pays back later as work that enters the artifact and matches the artist's intent. This pattern was not visible during the week. It surfaced only once the artifact was finished, reading what had entered it against each engineer's approach. We could read it at all because the work converged in one place, the shared artifact of Pattern 2, whose history made the comparison possible.



Ethnographic analysis

Context

Echo is a Cisco project, developed in partnership with the innovation agency H-FARM along with its media company SHADO, that brought five Cisco professionals (three software engineers and two user experience designers) into a one-week creative residency in Venice in March 2026. Working alongside the artist Paolo Fantin and the curator Giovanna Zabotti, and drawing on the musical grammar of composer Dardust, the team built an interactive installation for Venice Pavilion at the 61st Venice Art Biennale. Within that week the team created a prototype where AI interprets the written words of the visitors and transforms them into sound, light, and motion, within parameters the team and the artists defined. Each Cisco professional had a different background in using AI. Among the engineers, approaches ranged from extensive research before every prompt, to fast exploratory generation. The two designers brainstormed at a whiteboard and produced storyboards by hand, without using AI tools until the third day.

Observation #1

Research as input versus prompt as question
One engineer consistently treated AI code generation as the last step, not the first. On day 5, the team had to produce a ripple effect in the water. Before asking AI to generate anything, the engineer searched for visual references, consulted a scientific paper on Fourier methods for water simulation, and ran the same task across four different AI models to compare results. The resulting water reflections were realistic and close to what the artist had envisioned. Another engineer prompted exploratively, asking for four variations of the visualization without specific references. The results were visually generic: flat 2D waves with text.

INSIGHT #1

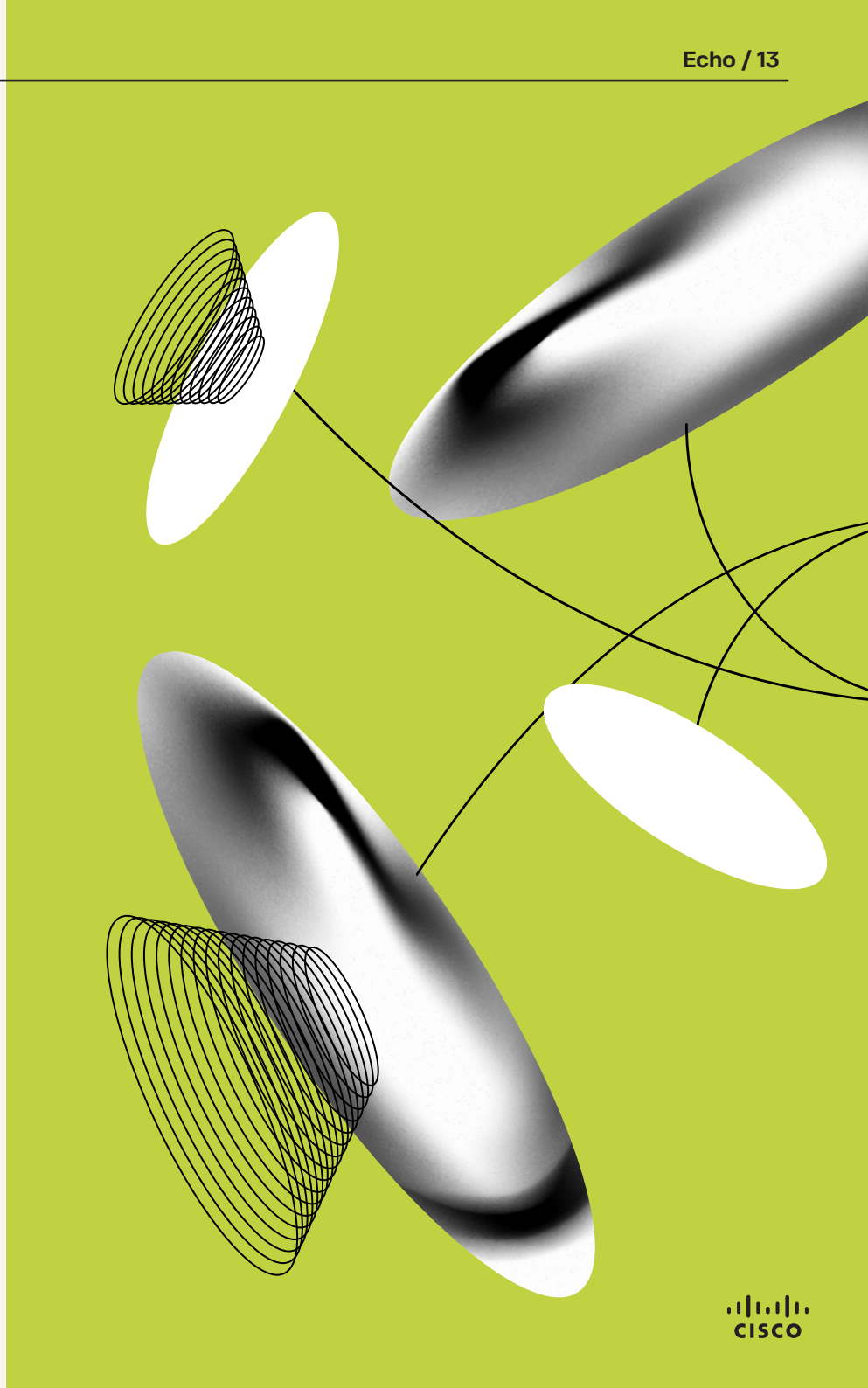
When the prompt carries references, constraints, and a visual target, the output gets closer to the artist's vision. Prompting with intention is a way to overcome the so-called AI slop. Otherwise, generative AI systems, no matter how sophisticated they are, tend to fill gaps in the instructions with their default knowledge, producing more options but none that match the project's needs.

Observation #2

Research develops project comprehension to capture project needs
On day 4, the team met with the artist and curator to review the work. One engineer proposed that visitors write their memories by hand rather than typing them, and described how the sound in the installation could guide people toward the writing stand. The idea placed a genuinely human gesture, handwriting, at the center of an AI-driven experience, so that visitors would encounter the technology without noticing it. The curator remarked that the proposal had an artistic sensibility not usually associated with an engineering role. This was the same engineer who had constantly researched across models and sources before prompting.

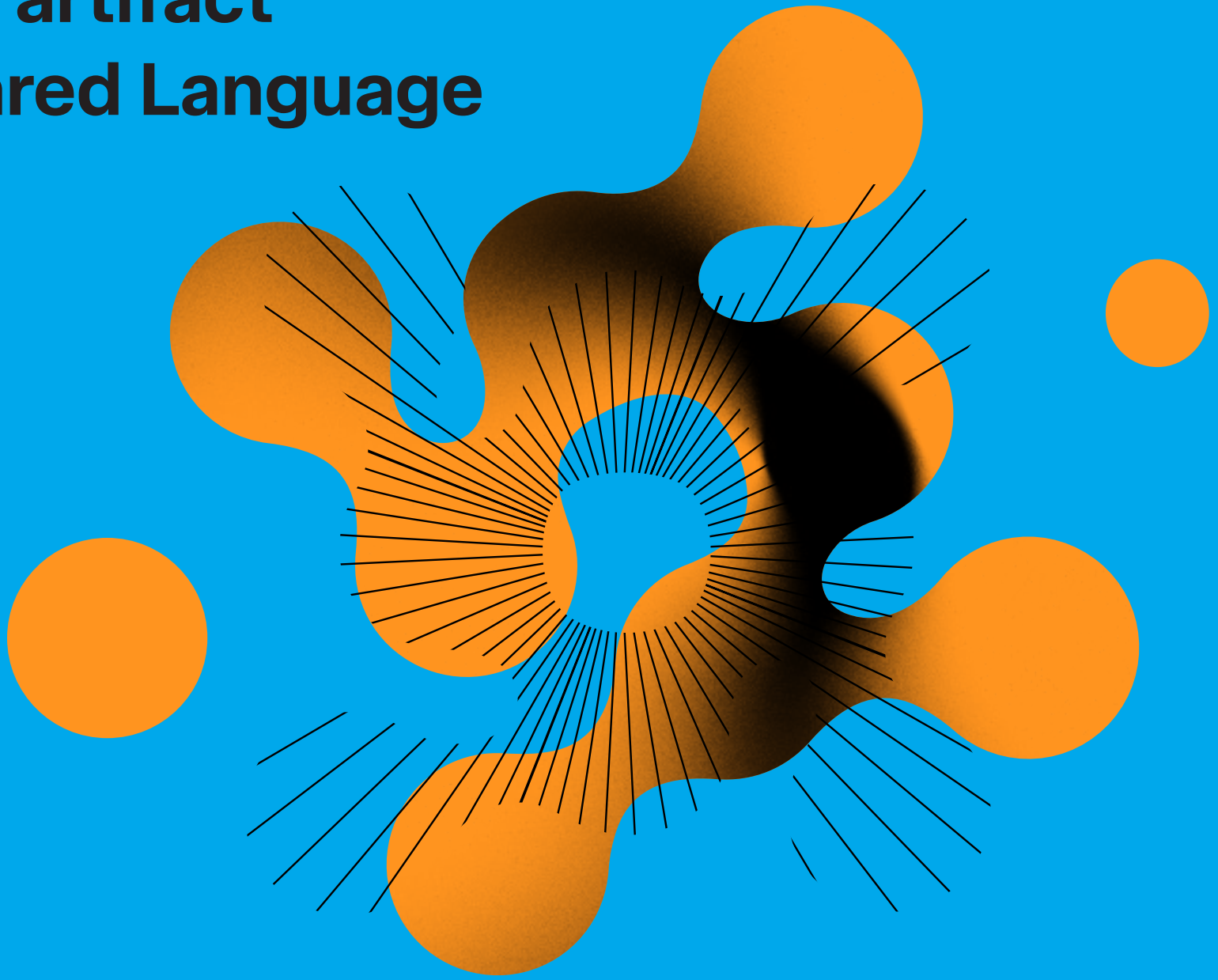
INSIGHT #2

Intentional prompting does not only improve AI output. It changes how a person participates in the project. The engineer who had researched across models and references could contribute beyond implementation: proposing ideas, discussing direction, and responding directly to the artist's feedback. This approach to prompting builds enough understanding of the project for the engineer to develop an independent perspective and capture the real intention of the artist and interpret it through a digital solution. This was a constant since this engineer had the highest percentage of interventions in the final artifact.



The AI artifact as Shared Language

2



WHAT IS IT?

With generative AI, teams can build a functional artifact from the start of a project rather than working only through representations such as wireframes, mockups, or prototypes.

Generative AI expands who can modify artifacts through prompting, beyond those who are able to write code.

Ideation therefore moves from generating ideas in the abstract to shaping them concretely. Team collaboration starts to resemble an artisan workshop: people work around the same piece rather than handing deliverables from one role to the next.

The artifact becomes a means of communication and decision-making: a shared and visible language that can evolve continuously. In meetings, the team can iterate on it, changing its shape and functions while seeing the results immediately.
The same interaction pattern applies to the client and other stakeholders, who can react to the artifact, play with it, or watch the team change it.

Because AI compresses the time to build and modify an artifact, every version of the final solution is the most complete possible at each stage. **Therefore an AI generated artifact is no longer a minimum viable product to scale later. It is a maximum valuable product: craftsmanship that grows through fast iteration.**

The Artifact as Shared Language

How to implement it

• Reach the artifact early[1]
Structure workflows to produce a working artifact as soon as feasible. The earlier the team has something functional to work on, the earlier alignment happens through the object itself. Reaching this point quickly depends on a proper working environment set-up (repositories, enabling tools, and knowledge), and the necessary and sufficient understanding of the brief and its context.

• Make it the center of every conversation
When the team discusses direction, the artifact should be open and modifiable in the room. Decisions happen on the object, not about the object.

• Let non-technical contributors modify it
AI enables non-technical contributors to modify a functional artifact through prompting. Rather than only providing feedback, they can be given access to modify the artifact directly, or to work on a backup copy.

• Make the artifact become the project record
The artifact and its history (commits, logs, versions) embody the current state of the project. This reduces the need for status updates, alignment meetings and project documentation.

Strategic implications

• From specialist methodology to shared literacy. Generative artifacts live in code, not in slides or documents, and AI compresses the time to write and modify them. Working in this way requires everyone involved in the project to become familiar with approaches drawn from software engineering practices, particularly Agile and Lean methodologies. This includes working with version control and CI/CD pipelines, which support continuous iteration and improvement. Within this framework, the whole team can contribute meaningfully to the artifact's evolution.

• From minimum viable to maximum valuable product. Because AI compresses build and iteration time, every version carries the full craftsmanship of the team at that stage. What reaches the client is not a minimum standardized product, but a full-fledged artifact that can be hyper-personalized to get the maximum result.

• Extract value from the project record.
The artifact is a living object: in a generative system, commits and messages are increasingly authored by the AI alongside the team. The history of versions, iterations, and attempts becomes an "active project archive" that can be analyzed and reused while the project is still going on and automatically feed lessons learned into new cycles. It holds the collective intelligence of the team and the AI together, in a usable form.

PITFALL MANAGEMENT

Risk #1. Artifact fixation. A functional artifact can quickly become the reference point for subsequent decisions. This is the risk of artifact fixation: because the artifact is tangible and functional, attention shifts from questioning the problem to refining the solution. The team may end up building the wrong thing efficiently, and the more developed the artifact becomes, the harder it is to discard and rethink. Keeping the artifact at the center of discussion helps ensure that it remains open to challenge as it evolves.

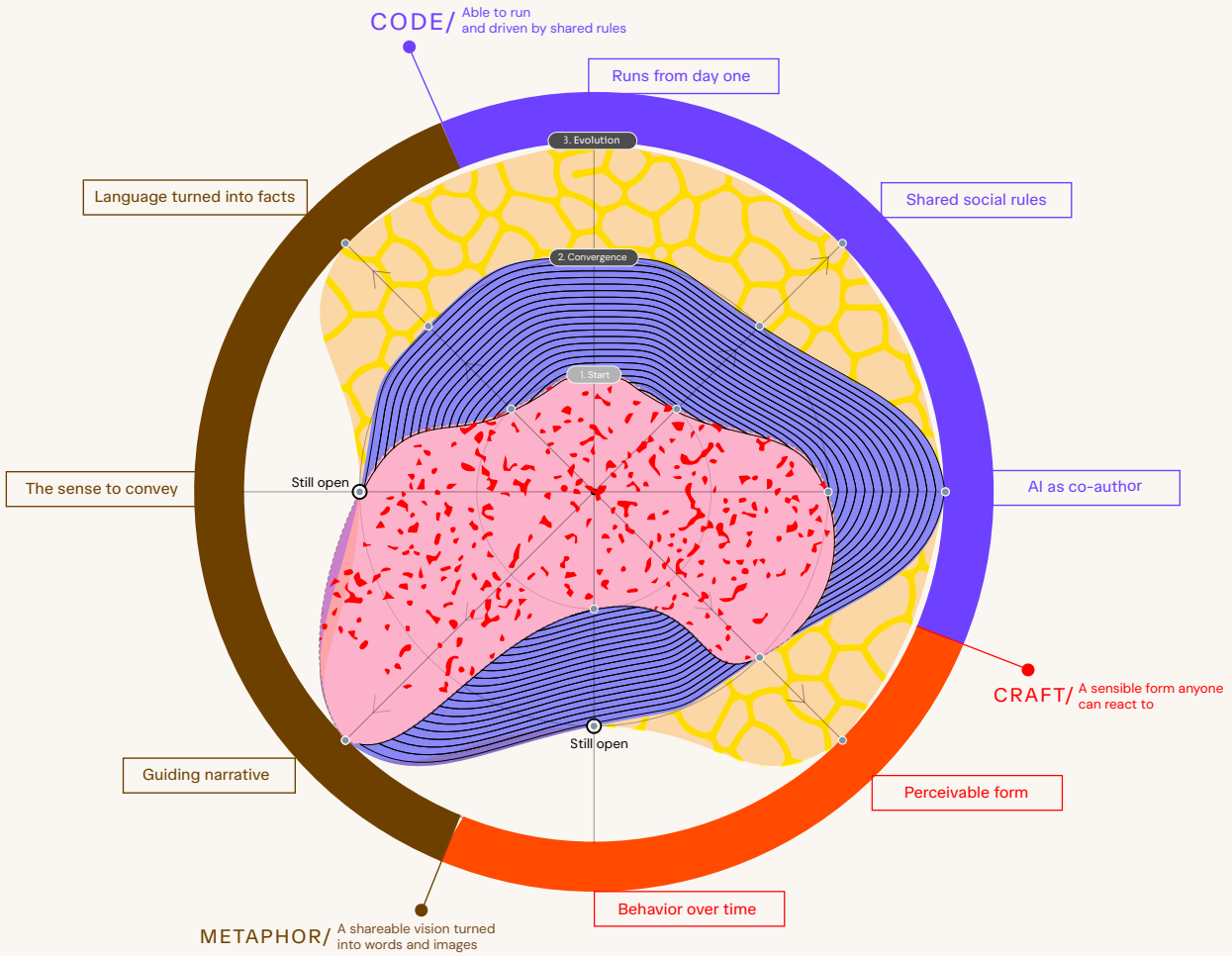
Data analysis / 2

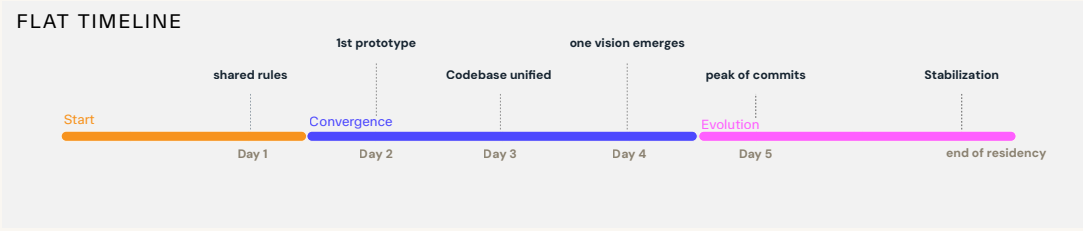
It isn't just code: the artifact's fundamental ingredients

The artifact is the product of a workshop dynamic, a collaboration that transcends roles. We identified three main ingredients that compose it: code, craft, and metaphor. The code makes it run and holds the rules that let several people work on it at once, so it runs from day one. Craft is the perceivable form, the surface anyone can read and ask to change without opening the code. Metaphor is the guiding image that gives the other two a direction and carries the sense the work should convey, here water as memory.

The rings inside are the three moments of the week, and they only grow: each keeps what the one before it added. The metaphor takes shape first and sets the work in motion before any code runs. As the team establishes shared rules for the repository, the artifact begins to run and the model commits code as a co-author, until it settles into a form people can see and react to.

Two axes never reach the edge. The parameters never stabilize, because they accumulated faster than the team could map what each one changed, and the sense the work should convey stays open, because translating the artist's language into the artifact was the hardest part of the week.

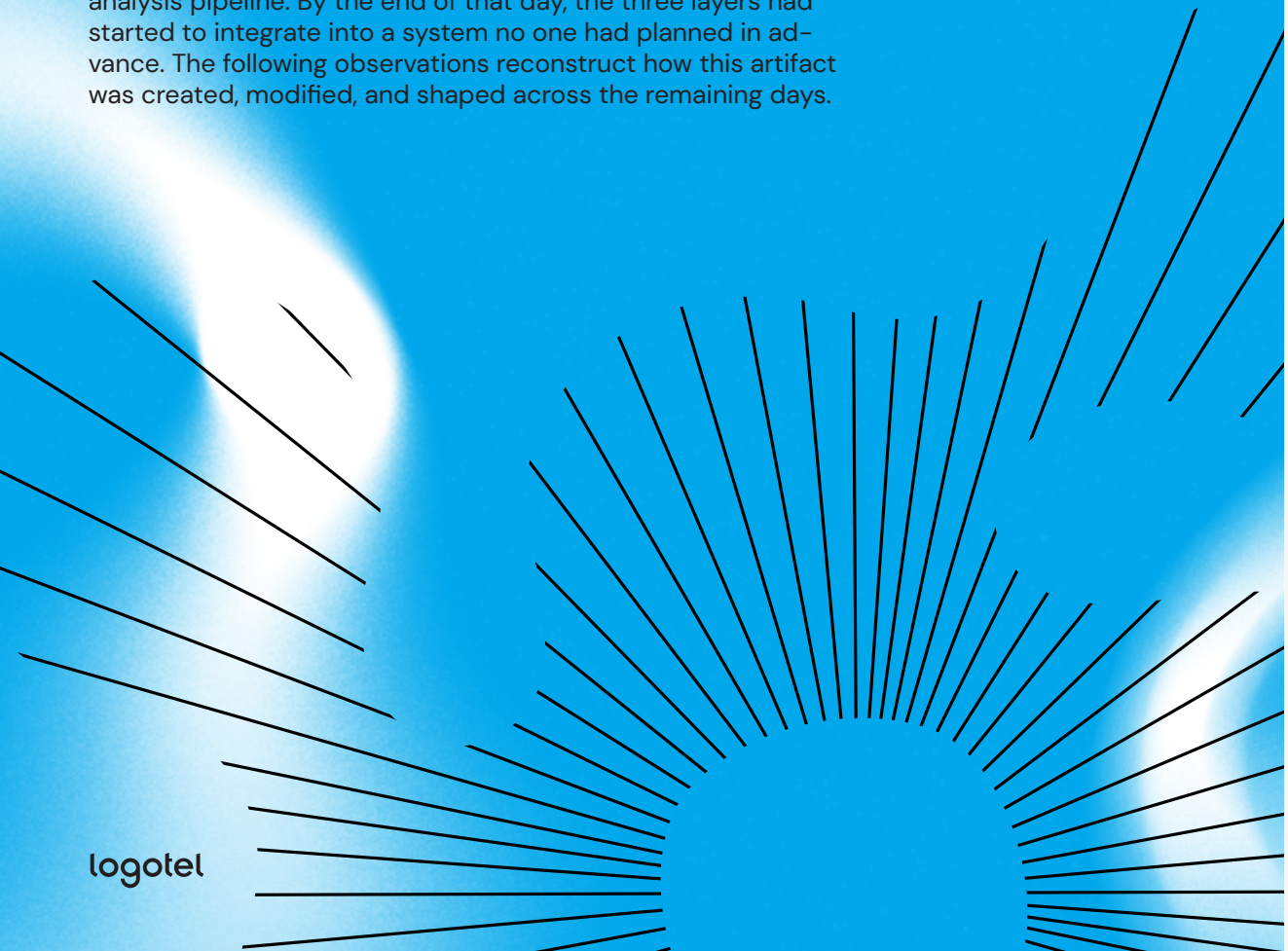




Ethnographic analysis

Context

For the first two days, each Cisco professional worked alone on their own direction based on the initial brief. On day 3, the team created a single shared project that would become the artifact of the immersive installation at the Biennale: one folder for the water rendering, one for audio and backend, one for the emotion analysis pipeline. By the end of that day, the three layers had started to integrate into a system no one had planned in advance. The following observations reconstruct how this artifact was created, modified, and shaped across the remaining days.



logotel

Observation #3

From individual experiments to one shared artifact
During the first two days, each engineer prototyped independently using multiple generative AIs: one built a water visualization, another generated audio from conversations, and another created an emotion analysis pipeline. In the afternoon of day 2, the artist reviewed the three prototypes and gave clear direction for unifying the different experiments into a single experience and artifact. The next day, the engineers merged their separate experiments into the shared codebase, each working in a different area of it. From that point onward, every modification happened on the same artifact.

INSIGHT #3

The artifact emerged from two movements.
The engineers were independently building prototypes that were addressing separate layers of the visitor experience. The artist saw them and defined how they should form a single experience. Neither alone would have produced the final architecture. At the same meeting, the curator saw the working prototypes and engaged immediately, recognizing artistic sensibility in the engineers' work. She had not reacted to the storyboards and user journeys previously presented. The artifact was something she could read and understand directly.

cisco

Observation #4

Bringing the designers into the artifact
On the morning of day 3, one of the engineers asked the designers what the visitor should see, feel, and when. The engineer needed this information to define the various user input-output relations in the end-user experience. The designers continued working on storyboards and visitor journeys at the whiteboard. The gap was not in willingness but in format: the engineers needed parameters, while the designers were producing narratives. And the designers did not yet have access to the artifact itself. On day 4, the same engineer taught them how to download a copy of the artifact and run it on their own devices. One designer managed to access it, manipulated the artifact, testing variations on a projector. Then they co-prompted, thought out loud, and explored combinations of visual parameters. One month after the residency, the same designer made her first commit.

INSIGHT #4

The artifact became a shared language for the designers only when they could run and modify it themselves. Once they could experiment with the artifact directly, their relationship to the project changed: from describing what they wanted, to seeing and testing it. But access alone did not close the gap. The speed at which the engineers iterated with AI remained faster than what the designers could contribute through their analysis and tools. The designer role is the one undergoing the deepest transformation, entering a process where the logic and the tools were different from what they knew. The first commit by a designer, a month later, is where that adaptation became visible.

Observation #5

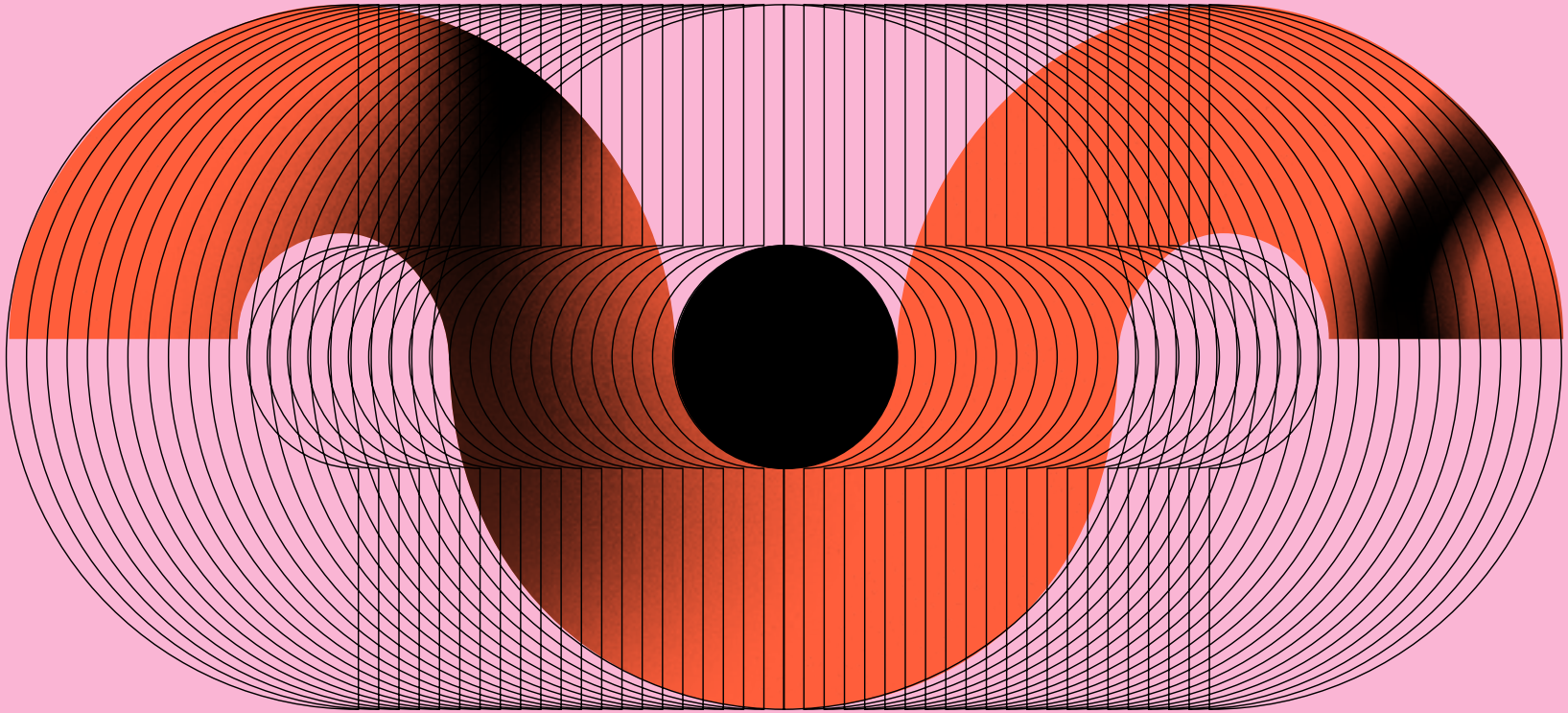
Direction and implementation in the same moment
On day 3, while the artist was refining the desired final section of the installation, one engineer began building it during the conversation. The engineer inserted a piano note that created a vibration in the water text rather than waiting for the discussion to end. The artist was struck by the speed but not yet satisfied with the result. On day 4, two engineers explored different implementations of the same effect in parallel within the shared project artifact and compared the results.

INSIGHT #5

When modification is fast and the artifact is shared, the distance between a request and its implementation collapses. Direction and construction happen at the same time, on the same object. This changes the shape of the design-driven collaboration, where divergence and convergence are no longer siloed phases of the process, but melt into a continuous process: the artist reacts to what appears, redirects it, and each feedback becomes the starting point for the next iteration. By working on a shared workspace, made of a mix of code, craftsmanship and meaning making metaphors, different attempts can be compared and selectively merged into the same artifact.

Designing the Governance Layer

3



WHAT IS IT?

The governance layer sits between human intervention and generative models.

We named it "layer" because it is an in-between enabler: it connects qualitative human intents with generative AI models.

It becomes increasingly important as a project needs to evolve and - at the same time - cannot tolerate regressions. So it takes the form of a visible interface, a sort of control panel that exposes the artifact's variables the team can modify, without requiring prompting to directly change the artifact.

Generative AI operates with a degree of agency that makes it both productive and unpredictable. Left unconstrained, that agency drifts: outputs diverge from intent, and the artifact loses coherence across successive modifications. Over-constrained, the system stops generating and becomes a template. The governance layer exists to hold this tension: it defines the boundaries within which the AI operates freely, without eliminating the variability that makes generative technology worth using.

To do this, the layer must achieve three things. **It must preserve coherence across modifications**, ensuring that changes in one part of the artifact do not break the integrity of the whole. **It must allow targeted, granular control**, so that teams can adjust specific aspects without unintended side effects. And **it must carry aesthetic and stylistic intent**, translating qualities of tone, feel, and expression into variables the system can apply consistently. These three functions correspond to three levels of control: context curation, precision, and expression.

Designing the Governance Layer

How to implement it

• Identify and expose the right parameters[1]
Each parameter surfaced is a point where human judgment enters the artifact. Expose too few and creative direction cannot reach it; expose too many and the layer becomes unnavigable. Technical roles know which parameters the system can expose; creative roles know which of those matter for the end-user experience.

• Define the boundaries for AI exploration
The degree of freedom assigned to each parameter depends on its function. Context curation parameters allow more room for the AI to combine elements coherently across the system. Precision parameters require a contained range to avoid unintended side effects. Expression parameters need the tightest boundaries, because here the AI must follow specific aesthetic intent rather than propose variations.

• Keep it usable
Each parameter should make clear what it changes in the artifact. If the layer accumulates too many variables without mapped relationships, it becomes difficult to navigate. As the project progresses, parameters that are no longer necessary should be removed.

Strategic implications

• Governable improvement. Once the governance layer, the control panel of the artifact is built, every subsequent evolution of the artifact is constrained by the same designed boundaries. Quality does not depend on individual judgment alone: the layer itself guarantees that changes stay within a coherent range.

• Feedback becomes parametric. When feedback reaches the artifact through its parameters, the artifact changes during the conversation. Traditional formats, mood boards, verbal notes, written specifications require someone to interpret and implement them after the session. That translation takes time and bears the risk of misalignment.

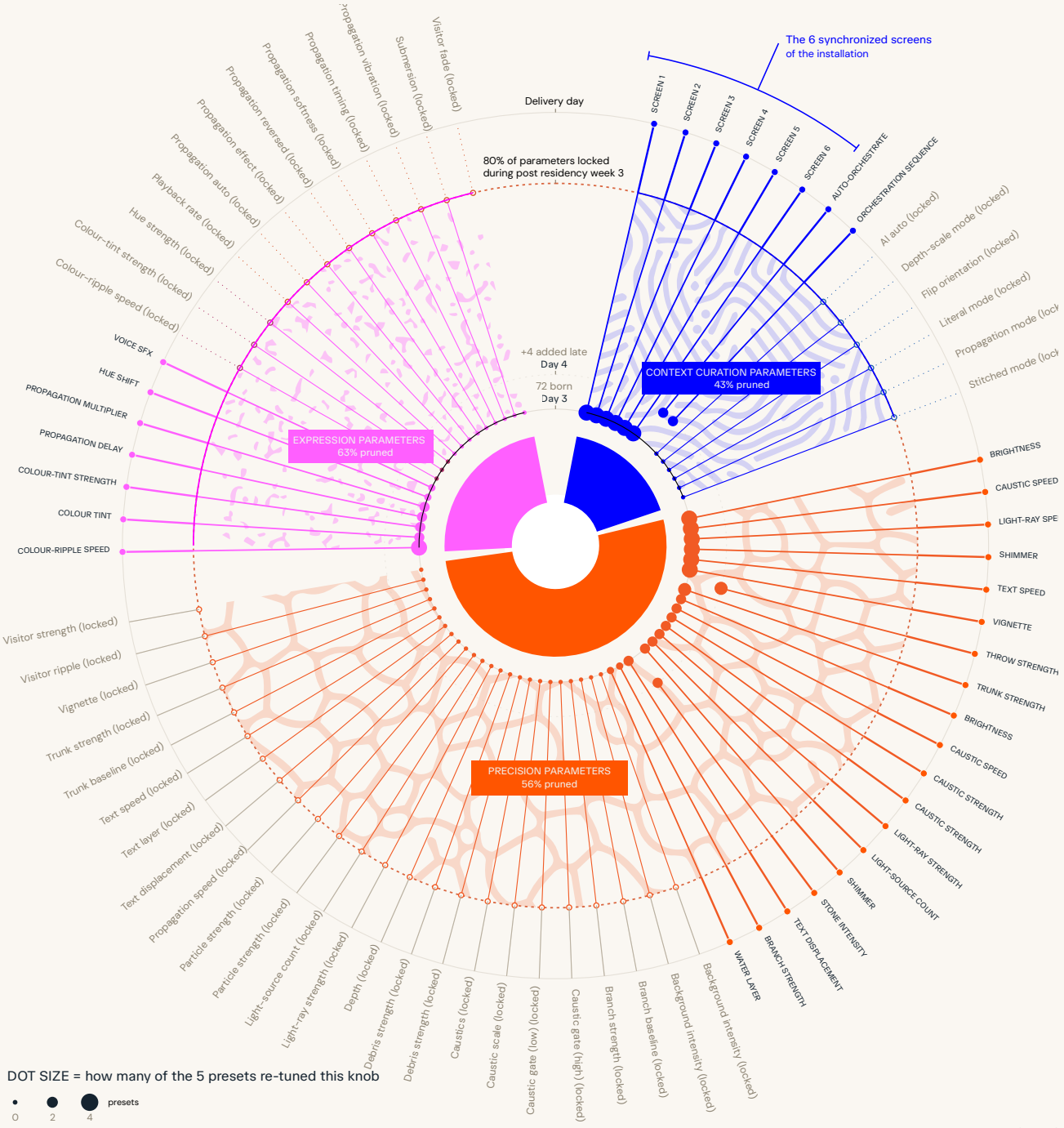
PITFALL MANAGEMENT

Risk #1. Arbitrary parameters. The parameters chosen here are not neutral technical decisions. If they are wrong, every subsequent iteration builds on that error without making it visible. Parameters that are never questioned stop being choices and start being constraints.

Data analysis / 3

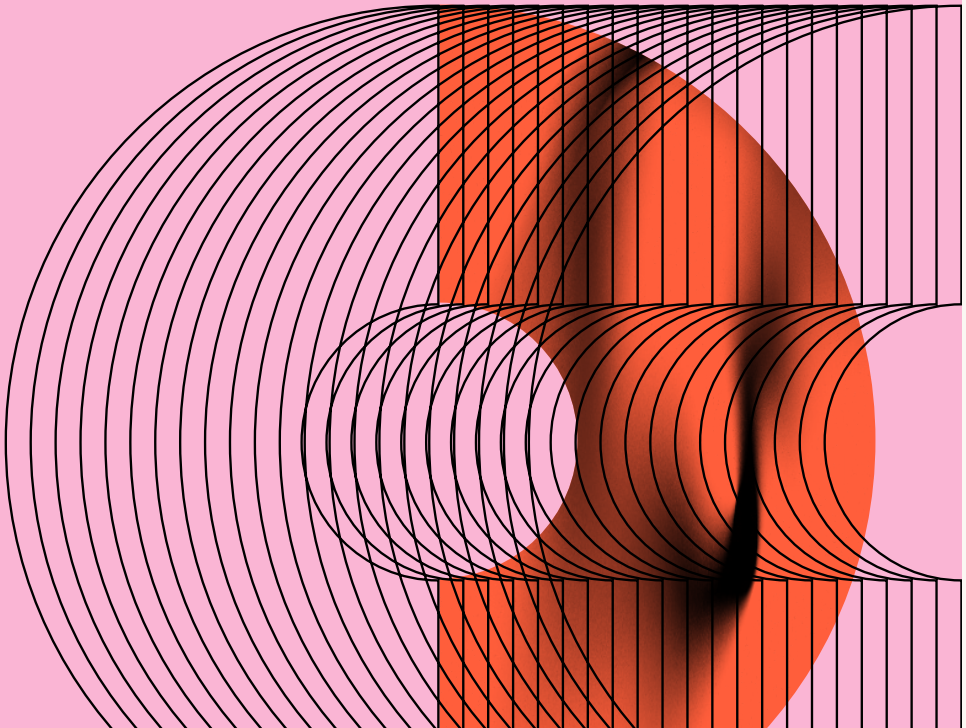
Finding the right balance between model's freedom and excessive control

The governance layer was born from the need to stabilize the artifact: to make it evolve toward the final demo in a controlled way.
These control dynamics are grouped into three families of parameters; in the visualization each ray is one parameter. Context Curation keeps the six screens of the artifact coherent and leaves the model more freedom. Precision parameters controls water, light, and text, the visible surface through which the artist's intent required the finest control, which is why it carries the most parameters. Expression controls tone, color, and emotion: the family where human judgment seemed hardest to delegate to the model.
A ray's length is a parameter's life. The count traces the residency's discussion: each request added a parameter as it came up, so the field grew to more than twice the number of parameters that reached Delivery Day. Two patterns explain the pruning. The first is trust: Expression was pruned most aggressively (-63%) because the team found the model could hold expressive coherence without a separate knob for every variable. Governing AI non-determinism does not require exhaustive control; it requires control in the right places. The second is stabilization: a parameter was removed when the decision it represented had been made. What remained adjustable were the parameters for which judgment was still open.



Ethnographic analysis

Context
The engineers built the installation artifact with generative AI; the artist directed it through periodic reviews using sensory language. The Cisco team, accustomed to technical and quantitative specifications, struggled to translate sensory language into changes on the artifact. The governance layer was built to close that gap by exposing the artifact's visual and audio parameters directly, it gave the artist a way to act on the artifact without going through interpretation. By day 5, the panel controlled dozens of variables across six synchronized screens. The following observations reconstruct how feedback reached the artifact and when it did not.



Observation #6

From conceptual description to machine instruction
The artist gave direction using evocative terms like "oneiric," "deep as solitude," "not figurative." The Cisco team struggled to turn these into machine specifications. To interpret his words, they used approximations like "mysterious" or "more intense" but didn't know how to translate them into the artifact. They realized they needed a layer that could translate the artist's direction into granular modifications on the artifact.

INSIGHT #6

Every verbal approximation introduced a margin of error between what the artist intended and what the team built. The control panel eliminated that margin: defining and using the parameters made artistic direction testable and adjustable directly on the artifact.

Observation #7

From descriptive material to grounded reaction
During day 2, the designers presented a storyboard of the visitor experience. The team listened, but the discussion did not continue and the storyboard did not shape the work. The same pattern repeated on day 5: a designer presented a mood board with color palettes and AI-generated renderings of water ripples, during a review with the artist. The team looked at it, but the references did not reach the artifact. Instead, when the engineering team showed the working artifact through the control panel, reactions immediately turned into adjustments: parameters shifted, directions changed, the artifact changed in the room following the stakeholder requests in a controlled manner.

INSIGHT #7

The storyboard and mood board remained external to the governance layer, and so they remained external to the artifact. Both are narrative formats: they describe intent but require someone to interpret and translate them into parameters after the session. The governance layer removes that translation step. It turns reactions into granular, bounded modifications, immediately visible on the artifact, which is why stakeholder requests generated direct adjustments instead of remaining observations.

Observation #8

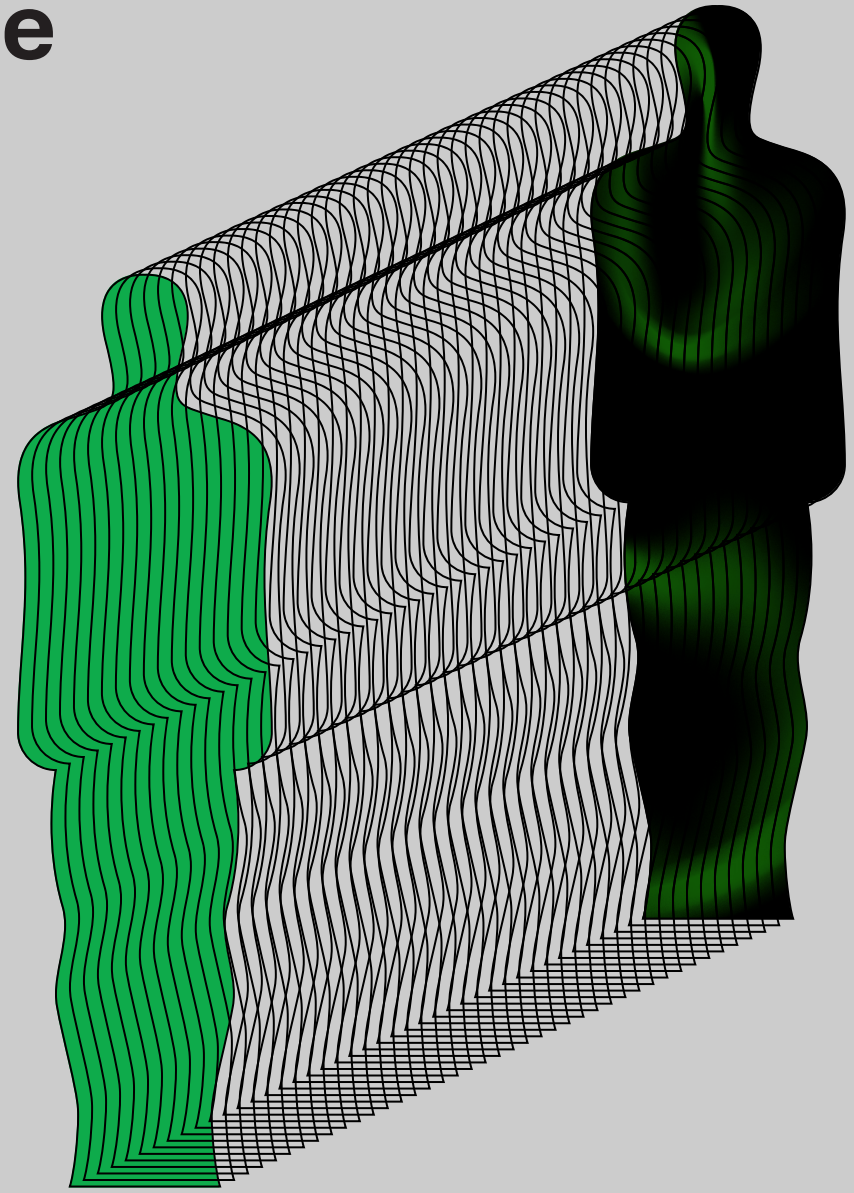
Parameter accumulation without mapped relationships
On day 5, the governance layer controlling the artifact outputs had become difficult to use. Parameters had been added throughout the week without clearly mapping which inputs affected which outputs. The project lead requested a simplification into a set of fixed preset states because reaching a specific visual result had become too difficult.

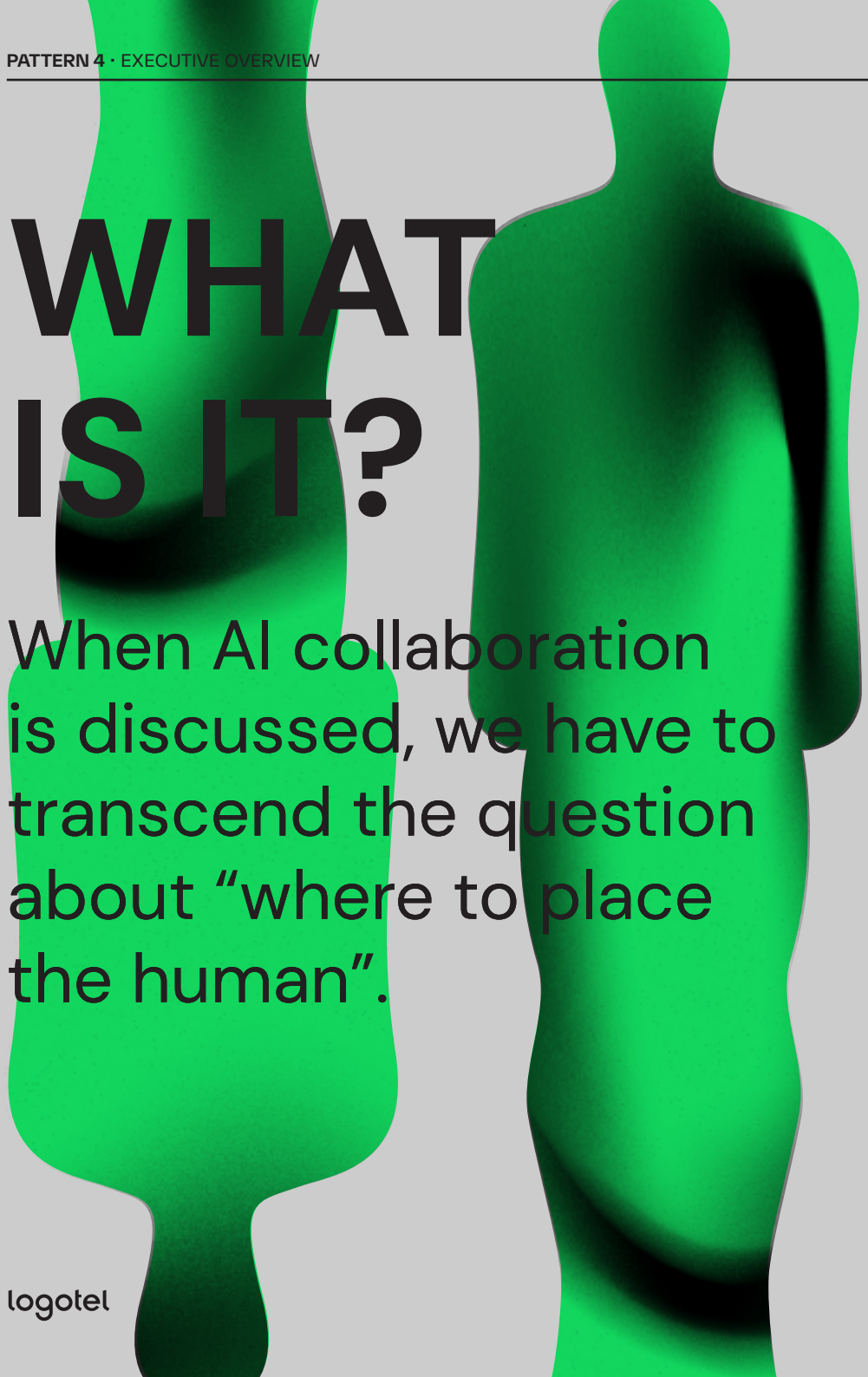
INSIGHT #8

A governance layer remains usable when each parameter is connected to a relevant effect on the artifact. The design challenge is deciding which relationships between input and output should remain adjustable because still valuable for the final user experience, and where the AI should be allowed to vary freely within those boundaries (parameters linked to expression and precision). The same applies with more technical parameters for the engineers (parameters linked to context and precision).

The Anchor Outside the (Agentic) Loop

4





WHAT IS IT?

When AI collaboration is discussed, we have to transcend the question about “where to place the human”.

The “human in the loop” framing fits an industrialized model of production, not creative work, where the human contribution is a point of view, the right problem to solve, an outside perspective.

The anchor outside the loop names this position: at the right distance from the agentic loop. The anchor is typically the client or a stakeholder with direct knowledge of the context the project operates in. They are not required to write prompts, run models, or modify the artifact directly. **What they bring is taste, perspective, and judgment grounded in a reality the model cannot access: the physical space, the audience, and the brief as it exists in the world.**

Generative AI foundation models make vast amounts of knowledge available, but that knowledge is generic. It has no direct relationship to the specific conditions of the project. The agentic loop, in which engineers, designers, and AI models iterate together, has a momentum of its own. It can generate variations that lead to dead ends, increasing token costs and time spent refining outputs without producing a satisfactory result.

The anchor breaks this momentum from outside the cycle, not from outside the project, and acts as a hook: a fixed point linked to reality that the loop cannot produce on its own. From this perspective, it can also be seen as a way to optimize production costs by prioritizing quality over quantity.

The Anchor outside the (Agentic) Loop

How to implement it

• Choose the right anchor[1]
The anchor is a hook in the agentic loop: a fixed point that the Generative AI cycle must adapt to rather than absorb. They bring a form of judgment that cannot be written into specifications or prompts: what feels right for the space, what resonates with the audience, what is coherent with the intention. Selecting someone for their taste, experience, and proximity to the real problem helps steer the project in the right direction through final delivery.

• Structure their interventions
The anchor can act in three ways: interrupting the loop to force evaluation of what exists, opening alternative directions that redirect the exploration, or validating the work against the real context. By defining when they intervene the overall quality of the outputs increases. It can happen at key milestones, when the team has produced a set number of variations, or at regular intervals.

• Feed their reactions back into the process
When the anchor reacts to the artifact, those reactions become constraints and inspirations for the next round of generation. Because the anchor stays outside the agentic loop, it is crucial to capture their feedback in the workspace so the team can act on it.

Strategic implication

• From production to discernment.
Without an outside anchor, the natural tendency is to keep generating variations. The anchor forces a different question: not "what else can we produce?" but "is this the right one?"

PITFALL MANAGEMENT

Risk #1. Unrecognized anchor. Assigning the role is necessary but not sufficient. Someone the team does not recognize as an authority of judgment gets ignored regardless of their title. Someone who can advise but not stop the process gets overridden by the momentum of generation. Someone placed in the role by seniority rather than by trust and credibility holds a position with no effect on the work, because the team has no reason to pause for their judgment.

Data analysis / 4

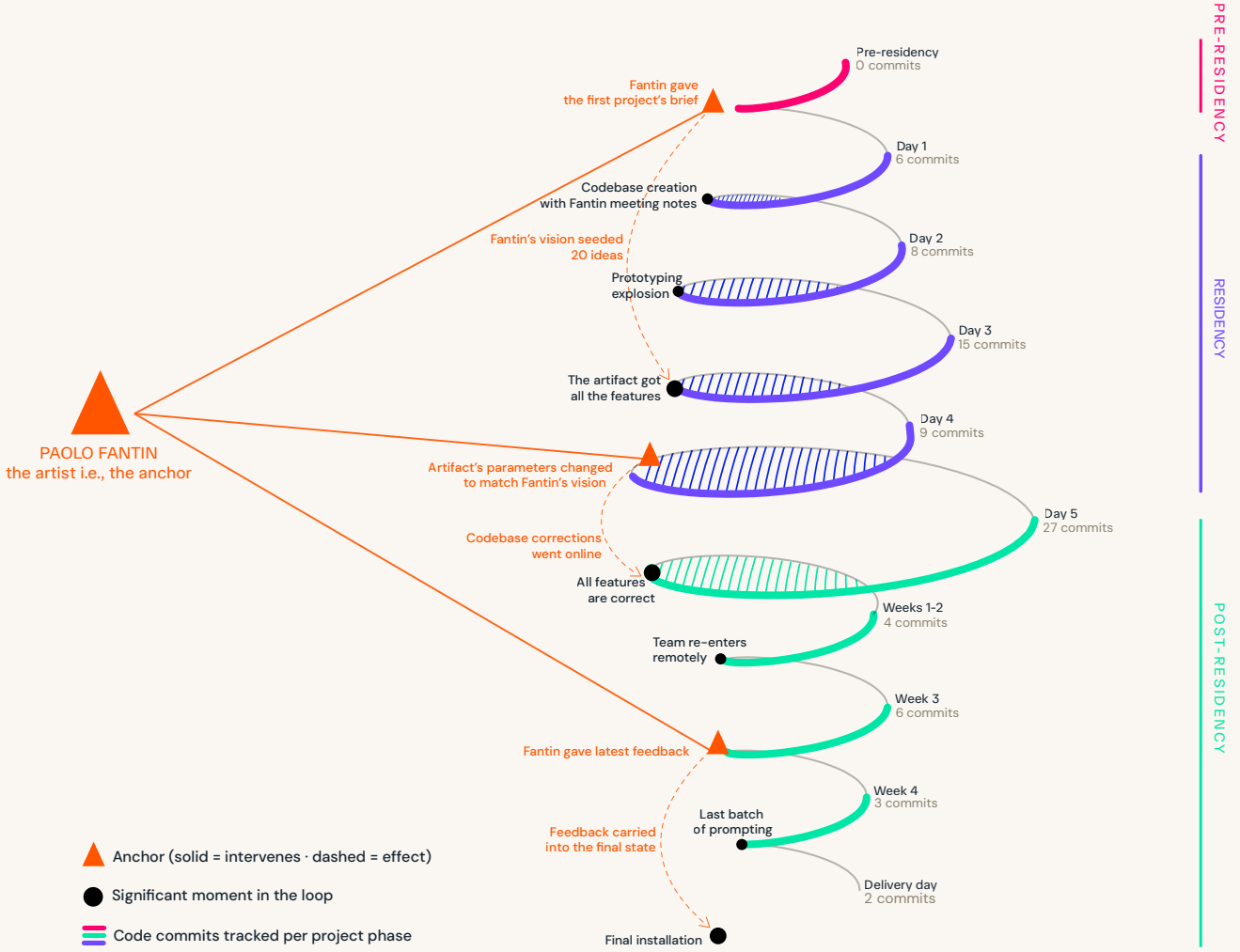
The correlation between the anchor's inspiration and code production

The anchor outside the loop directed this AI-native project toward its final demo. The artist's interventions generated code, discussions, and collaborations, expanding the creative dynamic productively. We correlated his interventions with code production to make the relationship visible.

The spiral is the agentic loop, one coil per day; its diameter is the commits made that day. The artist stays outside it: he neither commits code nor engages in prompt engineering. The solid red line is the moment he reaches in through reviews; a dashed red line is what that move produced inside the loop. Time runs from the pre-residency brief through the five days to delivery.

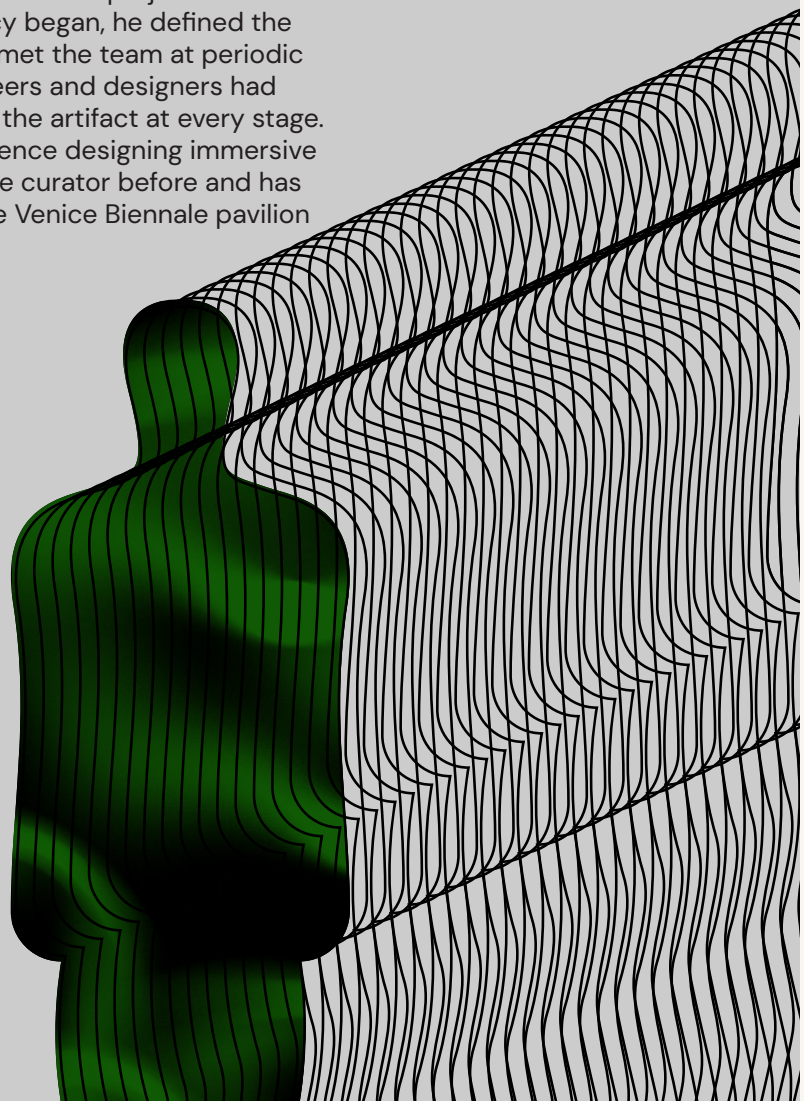
The artist produced none of the work, yet from outside he reshaped what the loop generated. Three moves: his pre-residency vision seeded the prototyping explosion; his feedback on the base color pulled the build back toward his intention, just before the busiest day; his preset of parameters carried the demo into its final state.

What the chart shows is the order: he intervenes, then work follows. His moves carried because of two decades of situated experience: the space, the audience, his earlier installations. Distance from the loop alone would not have done it.



Ethnographic analysis

Context
Paolo Fantin is the artist and scenographer working with the team. He did not use any AI tools during the residency. He was nevertheless deeply involved in the project's creative direction: before the residency began, he defined the metaphor of water as memory; he met the team at periodic reviews to react to what the engineers and designers had built; and his interventions shaped the artifact at every stage. He has over two decades of experience designing immersive installations, he has worked with the curator before and has worked in the physical space of the Venice Biennale pavilion before.



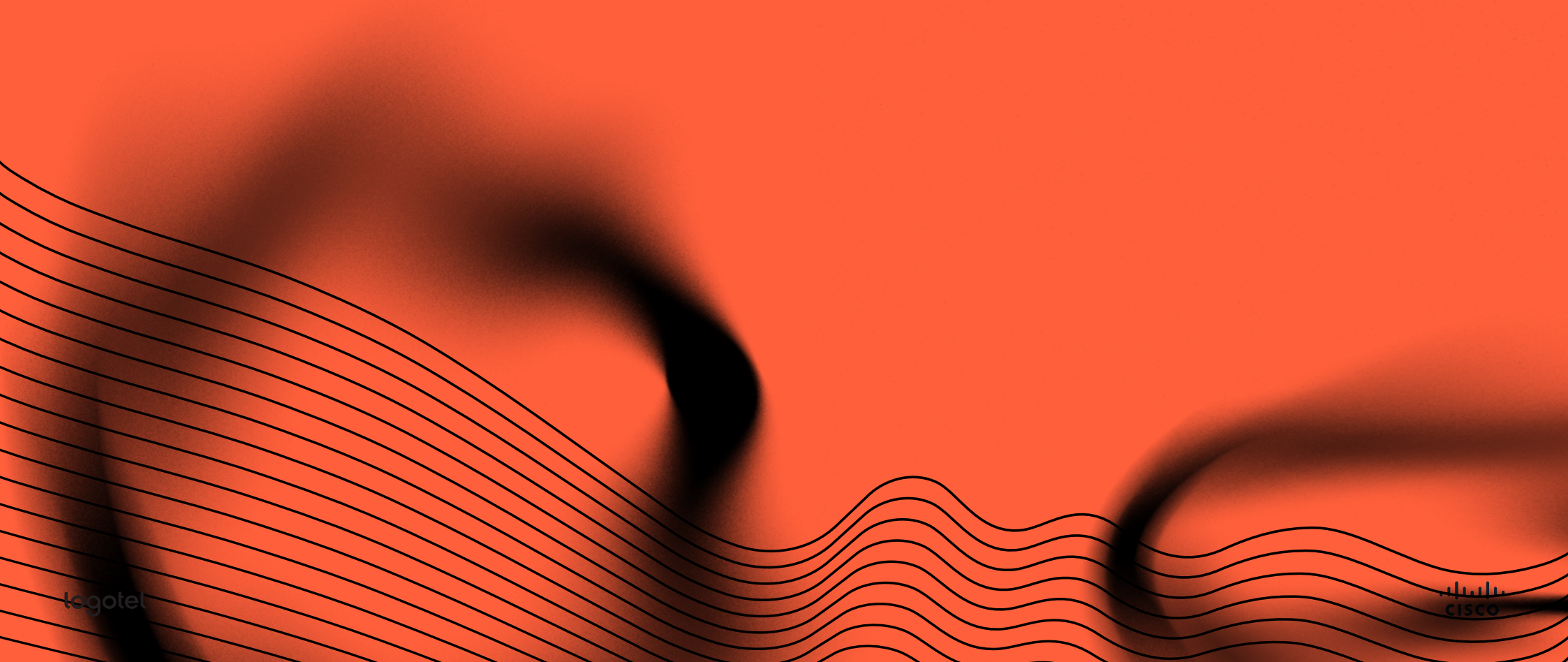
Observation #9

Fifteen prototypes and one person who did not generate
The three engineers produced more than fifteen iterations of different parts of the artifact over the course of the residency. The artist generated none of them directly, but shaped each through precise, situated direction. When the voice artifact was ready but uneven, he asked for a precise rhythm and beat it out with his hand: "tan tan tan." When the visual artifact was too bright, he asked for night blue, with the background words less legible at the final step. When the team had built three separate artifacts (sound, emotion pipeline, visualization), he triangulated them himself into a single experience with an idle state and an active state. To show how vibration should render on water, he played a video of one of his earlier installations as a reference.

INSIGHT #9

From inside the generative workflow, Paolo's interventions often appeared as interruptions: the engineering team kept producing variations, and he repeatedly stopped to reconsider, refine, or refuse the work. But the specificity he introduced did not come from the generative system itself: it came from his direct knowledge of the space, the audience, and what had or had not worked in his previous installations. His interventions shaped the work because they carried both authority and contextual experience.

Does Design Thinking Need an Upgrade?



Does design thinking need an upgrade?

The four patterns documented in this white paper point to broader and open questions: what happens to the structures that have organized creative work when generating and modifying artifacts becomes inexpensive and continuous?

For roughly two decades, design thinking has provided a common structure for organizing creative work. By dividing projects into phases and defining roles, methods, and handovers, it allowed multidisciplinary teams to coordinate their efforts around a shared process.

Entering the residency, we expected some version of this structure to organize the work. Instead, the project advanced under different conditions. The residency nevertheless produced a functioning artifact, suggesting that other forms of coordination had emerged in its place. The following sections examine three still open questions that surfaced repeatedly throughout the week: how is practice learned when work centers on a shared artifact? How does coordination happen when roles become fluid? And how is judgment exercised when generating new variations becomes easier than evaluating existing ones?

1. From Systematizing to Craft

During the residency, generative AI shifted collaboration beyond the phases and roles prescribed by design thinking. Design thinking's phases and roles exist to sequence work when building is slow and costly; once modifying the artifact became inexpensive and continuous, that sequencing lost its reason to exist. Generative AI was also not only a tool inside that process: it behaved as a generative party in its own right, producing variations no one had scripted, with a logic participants had to learn rather than command.

What held the work together was an emerging distributed craft. Rather than waiting for knowledge to be formally shared, participants developed it through imitation and curiosity. Neither is an efficient way to transfer information: both are creative acts, open-ended and not aimed at a specified outcome, closer to play than to instruction, which is exactly the kind of activity a staged process struggles to schedule for. Imitation involved reproducing others' actions across disciplinary boundaries: one engineer, for example, learned to sketch alongside designers and code alongside colleagues, eventually becoming a key connector within the team. Curiosity involved experimenting with tools to understand their behavior and recreate intended experiences, as another engineer did when translating an artist's direction into the work. In both cases, curiosity was directed as much at the model's own generative behavior, what it would and would not do, as at a colleague's. Both mechanisms required role fluidity: participants had to understand another person's practice well enough to act within it. This differs from design thinking's user-centered empathy. The relevant form of empathy here was collegial, the capacity to work inside a collaborator's logic, methods, and constraints. That collaborator was not always human: working inside a model's logic and constraints called on the same capacity.

The central question is therefore how this kind of live, cross-disciplinary apprenticeship could be cultivated on purpose, so that crossing into a collaborator's practice becomes a designed feature of how teams work rather than something that depends on who happens to be curious enough to attempt it.

2. In a Decentralized Process, Who Orchestrates?

In design thinking, the designer often held the work together through facilitation: aligning participants, structuring discussion, and moving the group through a shared sequence of phases. That facilitating role depended on the existence of a relatively stable process. Because generating is inexpensive enough for each person to iterate on their own, in parallel rather than in turn, no single shared review point remains for a facilitator to organize work around. When AI-mediated work no longer follows a common linear path, facilitation alone loses part of its object: there may be no single process to guide and no consensus to establish in advance. Facilitation does not disappear, however. It takes the form of mediation and translation. Both now answer to a third party that was not in the room before: the model's own generative loop, which keeps producing on its own momentum whether or not people are aligned yet.

Mediation is the work of bringing together divergent instances: visions, intentions, constraints that, left uncoordinated, produce prototypes with nothing in common. This urgency is specific to AI-native work: when each team member works through a private conversation with their own model, the outputs can diverge significantly without something holding them toward a shared direction. Divergence has two sources at once: people's differing intentions, and the generative range of the models they each work with, and mediation has to answer for both.

Translation operates differently, it is a technical capacity that runs in several directions at once: carrying a person's intention into a form the model can act on; making the model's output legible to a collaborator from a different discipline; and bridging two people whose starting languages are different enough that neither can immediately interpret what the other is producing.

Each crossing requires knowing what is missing: the right references, a shared vocabulary, the right question to ask the model. Design thinking coordinates by sequence, one phase closes before the next opens, and a single role often holds the handover; here coordination ran the other way, through overlap, with many small translations like these, rather than one orchestrator, keeping the work aligned while everyone acted on the artifact at once.
If the traditional designer's prerogative was to know the client, AI-native work requires that skill to be shared across the team. Put differently: where design thinking located this knowledge in a single role, AI-native work distributes it wherever someone, a person or a model, needs to turn an intention into a parameter the artifact can hold. Fittingly, no single profile able to hold both the strategic thread and the code at once ever emerged to take on that role, which is not a gap so much as evidence that the coordinating function had nowhere left to concentrate: it had already spread across the team and the artifact.

The open question is therefore how organizations can intentionally design for this kind of concurrent coordination: many small acts of mediation and translation running across people and the artifact at the same time, rather than treating it as a gap left behind by a missing orchestrator.

3. From Criteria to Taste

Design thinking evaluates work against a specifiable measure: the user's needs, tested by degrees across cycles of prototypes, and validated as a group. During the residency, each of these gave way. All three depended on iteration being slow enough to create distance: a rough draft to improve by degrees, time for a group to weigh in, a need external enough to test against. Inexpensive, continuous generation collapses that distance on all three counts at once. What decided a good result was no longer whether it served a stated need, but whether it had arrived at the shape the moment seemed to call for: a fit nobody had specified in advance, settled at once against a near-finished prototype rather than a rough draft, by whoever remained outside the generative loop.

This kind of judgment triggered on contact with the artifact rather than its description: the team did not react to a storyboard, only to the prototype running in front of them. A color system could be dropped on the grounds that it sat too far from the metaphor, a verdict nobody at the table contested, though no one could have written the rule behind it beforehand. The call belonged to one person, yet it settled the matter for everyone else, as if a different verdict would simply have been wrong rather than merely someone else's preference.

This sense of fit did not only close the door on finished work; at points it reached back into how the work was built. Before any objection had been raised, the team added extra controls to the governance layer for the sole purpose of absorbing a rejection that had not yet happened and could not yet be named, room built in for the moment the artist might simply say they didn't like what came out. Taste, anticipated rather than expressed, was already shaping what the system could produce, working less like a filter applied to finished output than like a constraint folded into the making of it. That anticipation was possible only because curiosity and imitation had already spread a working sense of the model's own range through the team, the same creative activation that made craft possible in the first place, not a separate faculty held only by whoever delivered the verdict. Taste, here, was not only the gate at the end; it was already a constraint at the start. None of this is a degraded version of design thinking. Craft, mediation, and taste are three faces of the same shift: a mode of collaboration built for a different cost structure, workshop-like rather than phased, running on synchrony and overlap rather than sequence, decentralized by default rather than by exception, and still early enough to be designed rather than merely observed.

The open question is therefore how this kind of immediate, embedded judgment, taste that keeps pace with generation instead of following it in a separate evaluative phase, could be built into how teams work on purpose, rather than left to whoever happens to stand outside the loop.