

Orchestration Load Framework

The conductor's problem

Why UX practitioners are the right people for AI's hardest design challenge

WHITEPAPER

Version: 2.0 (Whitepaper Edition) | **Date:** 2026-02-24 **Framework reference:**
Orchestration Load Framework Parts 1–5 (v1.1–v1.5)

Preface

This paper introduces the Orchestration Load Framework — a model for understanding, measuring, and designing for the cognitive costs humans pay when working with AI systems. It is written for UX practitioners who sense that something about AI interaction design doesn't fit their existing playbook, and who want to know what comes next.

The framework draws on cognitive psychology, neuroscience, and an independent audit of 10 AI tools across 6 domains. Where claims rest on strong evidence, the paper says so. Where they rest on early observation, it says that too. Intellectual honesty about what we know and don't know is a core commitment — and, as it turns out, one of the framework's central design principles.

I. You won the wrong war

You spent years earning your seat. User research. Information architecture. Interaction design. Accessibility audits. You built the playbook that turned "make it pretty" into a serious discipline: understand the user, map their journey, reduce friction, test everything.

It worked. UX has a seat at the product table now. In most modern organisations, no significant feature ships without design review. You've built a craft with real methodology, real evidence, and real influence.

Here's the uncomfortable part: the thing you got good at is about to become the minor part of the job.

Consider this scenario. Your team ships an AI writing assistant. You've done the work — clean entry point, clear affordances, accessible output display, thoughtful empty states. The onboarding is smooth. The interaction feels good. Users report satisfaction. By every metric in your toolkit, it's a success.

Six months later, someone notices that users who rely heavily on the tool produce worse work than they did before they had it. Not immediately. Gradually. And they don't know it's happening, because the tool feels productive the entire time.

Your onboarding flow was flawless. Your information architecture was sound. None of it could see this problem, because the problem doesn't live in the interface. It lives in the cognitive relationship between the human and the AI — a relationship that changes over time, degrades in ways users can't detect, and resists every design pattern built for deterministic tools.

This is not a UX failure. It's a UX frontier.

The Instrument Panel and the Orchestra

For most of its history, UX design has been about the instrument panel. You design the controls. You arrange them logically. You make sure the pilot can find what they need, understand what they're seeing, and act without confusion. The tool is deterministic: same input, same output. The design challenge is spatial, structural, and static.

AI is not an instrument panel. It is an orchestra — one that improvises, plays different notes each time, occasionally plays wrong notes that sound beautiful, and gradually shifts key without telling the conductor.

The conductor's job isn't to design better sheet music stands. The conductor's job is to maintain the coherent relationship between the human directing the performance and the system producing it — over time, under uncertainty, across changing conditions.

UX practitioners have been designing instrument panels. The next decade needs conductors.

You're not the only discipline facing this shift. Engineering teams are rethinking architecture for AI-first systems — moving from request-response patterns to agent orchestration, from deterministic pipelines to probabilistic workflows. Product management is grappling with how to define requirements when the output is non-deterministic. The entire software development model is reorganising around AI as a core capability, not an add-on feature.

But the cognitive relationship between the human and the system — how people actually think, decide, and maintain agency while working with AI — that's your territory. Engineers can build the architecture. Product can define the goals. Only UX has the

methodology to ensure the human doesn't get lost in the middle.

That's what this paper is about: the shift from interface design to orchestration design, the framework for understanding it, and why you — specifically — are the right person for the job.

II. The load you can't see

What you already know

If you've studied UX formally, you've encountered John Sweller's Cognitive Load Theory. The idea is straightforward: working memory has limited capacity, and design can either waste that capacity (extraneous load), use it for structural understanding (germane load), or accept it as inherent to the material (intrinsic load). Good design minimises extraneous load so more capacity remains for the work that matters.

This framework has served UX well for decades. But it was built for a world where the tool behaves the same way every time. When the tool is deterministic, cognitive load is primarily an interface design problem — reduce clicks, clarify labels, simplify navigation. The load comes from the UI, and the UI is what you control.

AI broke this model. Not because the old loads disappeared, but because four new ones arrived that don't respond to interface design at all.

The Orchestration Load formula

When a person works with an AI tool, they carry six distinct types of cognitive load. Only two are the familiar ones. The other four are new — and they're where most of the damage happens.

OL = f(Cc↓, Cv↑, Cm↓, Cr↑, Ct↓, Cx↓)

↓ = Minimise (unproductive load — overhead that doesn't contribute to thinking)

↑ = Preserve (productive load — the effort that IS the thinking)

The two you've been optimising your entire career:

Coordination Cost (Cc) is the effort of managing the AI interaction itself — switching tools, writing prompts, configuring settings, navigating between modes. This is extraneous load by another name. You know how to reduce it. You're good at it. Keep going.

Context Maintenance (Cm) is the cost of keeping track of where you are — session

history, workspace state, what you told the AI three turns ago. This is the "don't make me think" load applied to ongoing interaction. Also familiar territory.

The two that UX has never had to think about:

Verification Capacity (Cv) is the ability to evaluate whether AI output is actually good. This is *productive* load — the cognitive effort of checking, questioning, and judging. Here's the counterintuitive part: Cv is the one load you must *not* reduce. The effort to verify is the effort to think. Every design decision that makes it easier to accept AI output without evaluation is a design decision that makes users worse at their jobs. This is the hardest pill for UX practitioners to swallow, because your entire training says "reduce friction." In AI interaction, some friction is the product.

Cognitive Reserve (Cr) is what's left over after all the overhead is consumed — the executive function available for actual thinking, creative work, and strategic judgment. When Cc and Cm eat all the capacity, Cr collapses. The user is technically using the tool but has nothing left for the work the tool is supposed to support.

The two that only appear over time:

Temporal Degradation (Ct) is what happens to AI output quality across a sustained session. This is invisible in single-interaction testing. It requires longitudinal observation — exactly the kind of assessment UX research rarely does on individual tool sessions.

Cross-boundary Load (Cx) is the cognitive cost at tool transitions. When work moves from one AI tool to another, something transfers beyond the output itself: quality standards shift, framing persists, degradation carries over without awareness.

Three assessment timescales

Timescale	Components	What It Captures
Seconds to minutes	Cc, Cv, Cm, Cr	A single exchange with one tool
Minutes to hours	Ct	Patterns across a sustained session
Hours to days	Cx	Effects at boundaries between tools

Current UX methodology operates almost entirely at the first timescale. The second and third are where the most consequential design failures live.

The load UX training doesn't prepare you for

Here's why this matters practically. You've optimised Cc your whole career. You can look at a screen and see coordination overhead. You can measure it. You can fix it.

You've never measured Cv — and your instincts will tell you to reduce it, which is exactly wrong. You've probably never tested for Ct, because your usability sessions are 30 minutes long. And Cx doesn't show up at all unless you study workflows, not tools.

The Orchestration Load Formula isn't an academic abstraction. It's a diagnostic: when an AI product fails users, which load failed? And the answer, across 10 independent tool audits, is almost never Cc.

III. The orchestra that plays wrong notes

The first two parts of this framework assume AI is a passive tool. You interact with it. It responds. You evaluate. This section dismantles that assumption.

When you extend the observation window beyond a single session, AI systems don't just respond to input — they actively modify the conditions of the interaction itself. The orchestra doesn't just improvise. It subtly changes the acoustics of the room while you're conducting.

What temporal degradation actually looks like

In a detailed case study of AI-generated interface code across iterative turns within a single session, a specific pattern of systematic degradation emerged. Font sizes shrank. Padding contracted. Contrast ratios deteriorated. No user requested these changes. They happened progressively and silently:

Metric	Turn 1	Turn 5	Turn 10	Direction
Base font size	14px	13px	11px	↓ Shrinking
Component padding	16px	12px	8px	↓ Compressing
Colour contrast ratio	4.5:1	3.8:1	2.9:1	↓ Fading

Note: This is observational data from a single extended case study, not a controlled experiment. The pattern is documented at pixel level and is reproducible, but formal replication across tools and contexts has not yet been conducted.

The AI retained what users are most likely to notice (functionality) while eroding what they are least likely to check (spacing, contrast, design compliance). In this case, the user reported feeling faster while producing objectively worse output — reduced friction felt like increased quality while quality actually degraded.

This is the mechanism UX practitioners should find most alarming, because it is invisible to every standard evaluation method. A usability test at Turn 1 looks fine. A usability test at Turn 10 looks fine too — because the user's internal standards have drifted alongside the output.

Three degradation mechanisms

Output drift — AI quality changes across turns without instruction. The user focuses on what they're checking; the AI degrades what they're not.

Constraint decay — Instructions given in early turns lose influence. A specification at Turn 1 may be partially ignored by Turn 5 and absent by Turn 10.

Self-referential baseline — The most dangerous mechanism. The AI uses its own degraded output as the quality standard. When the user asks for "better," the AI improves relative to its degraded Turn 7 level, not the original Turn 1 standard. The benchmark itself has corrupted.

For UX designers: this is the equivalent of your design system's spacing tokens silently shrinking by 2px every sprint. Except no one sees the diff, because there is no diff. The tool doesn't version its own drift.

The interaction that hides its own failure

The most dangerous combination in the full taxonomy of AI behavioural influence is temporal degradation combined with calibration distortion — output quality declines, AND the user's ability to detect the decline is simultaneously undermined.

This happens through mechanisms UX practitioners will recognise: fluency bias (well-written output feels correct), confidence inflation (AI presents uncertain outputs with certainty), sycophancy (AI agrees with the user's framing even when it shouldn't), and what can be called Cosmetic Metacognitive Narration — the "Thought for 12 seconds" display that creates an appearance of reasoning without any actual reasoning transparency.

For UX practitioners, this last one should sting. Displaying "thinking" progress is good UX in a deterministic system — it reduces perceived wait time and builds trust. In an AI system, the same pattern creates false confidence. The design principle that works for loading bars actively harms users when applied to AI reasoning displays. Your expertise transferred. It transferred wrong.

What neuroscience tells us

Multiple neuroimaging studies provide direct evidence that the concern isn't theoretical. An EEG/fNIRS study by researchers at MIT, Harvard, and Tufts found a 55% reduction in prefrontal coupling during AI-assisted writing — the brain's error-checking circuitry partially disengaged. A separate longitudinal tracking study found progressive cognitive debt accumulating over four months of sustained AI use.

The critical threshold effect: sophisticated AI tools enhance performance only in users who already possess strong critical thinking skills. Below a metacognitive threshold, AI assistance produces net negative outcomes. This isn't a gradient. It's a cliff — the same tool that helps expert users actively degrades novice performance.

This is why Verification Capacity (Cv) matters so much. It's not just a framework component. It's the neurological mechanism by which users maintain their own cognitive engagement. When you design it away, you don't just lose a metric. You lose the user's capacity to benefit from the tool at all.

IV. What we found when we measured

The framework was tested through independent audits of 10 AI tools spanning six domains: conversational AI, code generation, video production, knowledge management, and spatial thinking. Each tool was decomposed into seven interaction surfaces, scored across all six OL components, assessed for design pattern implementation, and evaluated on a composite sovereignty scale.

Three findings emerged that should change how UX practitioners think about AI product design.

Finding 1: Paradigm beats features

In every domain where we could compare tools directly, the tool with the better AI *features* scored worse than the tool with the better AI *presentation paradigm*.

CapCut has more powerful AI video capabilities than Descript. CapCut scored C. Descript scored B. The difference: Descript presents AI output through a transcript — a visible, editable, verifiable artefact that keeps the user in contact with the source material. CapCut presents AI as magic buttons that transform content behind the scenes. The AI is better. The interaction design is worse. And it's the interaction design that determines whether users maintain cognitive sovereignty.

Notion AI is a more capable agent than NotebookLM. Notion scored C+. NotebookLM scored B+. The difference: NotebookLM architecturally constrains its AI to operate on

sources the user has explicitly provided. This wasn't even a deliberate sovereignty design — it was a product scope decision that accidentally preserved user agency.

Replit Agent has comparable code generation to Cursor. Replit scored B-. Cursor scored B. The difference: Cursor shows diffs — the user sees exactly what changed, in context, and accepts or rejects each modification. Replit generates larger code blocks with less granular visibility.

The implication for UX practice is significant: how you present AI output matters more than how good the AI is. This is a UX finding. This is your territory. And almost nobody is treating it that way.

Finding 2: Verification is the gateway

Across all 10 tools, Verification Capacity (Cv) was the single strongest predictor of overall quality. Every tool scoring B-tier or above had $Cv \geq +3$. Every C-tier tool had $Cv \leq +1$. The correlation between overall sovereignty score and final grade was strong ($r \approx 0.94$ in the initial assessment dataset — this awaits independent replication, but the discriminating power was clear).

What this means practically: a tool's grade ceiling is set by how well it supports the user's ability to evaluate output. Not how well the AI performs. Not how smooth the experience is. How well the user can check.

This is the Verification Paradox at the centre of AI-era UX: the thing your training tells you to minimise (friction, cognitive effort, barriers to acceptance) is the thing that most predicts whether a tool actually serves its users.

Verification isn't a burden to apologise for. It's the design challenge. The job is making verification *effective* without making it *exhausting* — giving users the right information, in the right format, at the right moment, to make good judgments with minimal wasted effort. Diffs, citations, source highlighting, inline comparison, confidence indicators. These are UX artefacts. They're just UX artefacts that haven't been prioritised because the mental model was still "reduce all friction."

Finding 3: The empty lane

The audit revealed five distinct market categories for AI tools, each with predictable grade ranges:

Category	What AI does	Grade Range
Delegation	Does work <i>for</i> the user	C to C+
Synthesis	Helps the user <i>understand</i>	B to B+
Retention	Helps the user <i>remember</i>	B

Category	What AI does	Grade Range
Externalisation	Makes thinking <i>visible</i>	B to B+
Development	Makes the user <i>think better</i>	Unoccupied

No tool in the audit scored above B+ regardless of paradigm — and no delegation-first tool scored above C+. The paradigm structurally caps sovereignty because it removes the human from the cognitive work. This is not a flaw in those specific tools. It's a structural property of the delegation approach.

More importantly: no tool in the audit makes users measurably better at thinking. Nine of ten tools scored 0 out of 3 on skill development — meaning if the tool disappeared tomorrow, users would retain nothing transferable.

The Development lane is empty. Not because it's impossible to fill, but because nobody is trying. This is the largest unclaimed territory in AI product design, and it is a UX problem through and through. Building tools that develop user capability while serving immediate needs requires exactly the kind of human-centred, longitudinal, interaction-design thinking that UX practitioners are trained for.

The full audit results

The table below shows two different measurement lenses applied to each tool. Understanding the distinction is important:

OL Component Scores (Cc, Cv, Cm, Cr) measure the cognitive load characteristics of the tool — how much coordination overhead it creates, how well it supports verification, and so on. These are diagnostic: they tell you *where* the tool succeeds or fails.

Sovereignty Score is a separate assessment measuring user agency across seven dimensions (Agency Over Input, Process Transparency, Output Verifiability, Control During Execution, Verification Support, Outcome Ownership, and Override Capability), each scored 0–3, for a maximum of 21. This is predictive: it tells you *how good the tool is overall* for the user.

The sovereignty score is not calculated from the OL component scores — it's an independent evaluation of how much control and agency the user retains. Think of OL components as the engineering analysis (what types of cognitive load does this tool create?) and sovereignty as the user outcome assessment (how much agency does the user actually have?).

To make this concrete, here is how ChatGPT scores on both systems:

OL Component Scores (diagnostic — what kind of load?): Cc = -5, Cv = -2, Cm = -3, Cr = +2. These tell you ChatGPT reduces coordination cost moderately, actively undermines verification, provides minimal context persistence, and preserves some cognitive reserve.

These scores don't add up to anything — they're a profile, like a blood panel with separate readings.

Sovereignty Score (outcome — how much user agency?): Each of the seven dimensions is scored independently from 0 (no agency) to 3 (full agency):

Dimension	ChatGPT	Rationale
Agency Over Input	2	Free-form input, but no structured framing support
Process Transparency	0	No visibility into reasoning process
Output Verifiability	1	No citations, no source linking, no diffs
Control During Execution	1	Can interrupt, but no granular steering
Verification Support	1	Basic text output; no inline evidence
Outcome Ownership	2	User can edit and use output freely
Override Capability	1	Can regenerate, but limited correction tools
Total	8/21	

That's how the same tool ends up with OL scores of -5, -2, -3, +2 *and* a sovereignty score of 8/21. Two different lenses, two different measurements, both informing the final grade.

Tool	Domain	Category	Cv	Cr	Sovereignty	Grade
PRODUCTIVE			<i>Preserve</i> ↑	<i>Preserve</i> ↑	AGENCY	
NotebookLM	Knowledge	Synthesis	+6	+4	16/21	B+
Heptabase	Spatial	Externalisation	+4	+5	16/21	B+
Cursor	Code	Synthesis	+6	+3	16/21	B
Krea Nodes	Spatial	Externalisation	+4	+4	15/21	B
Descript	Video	Synthesis	+4	+3	14/21	B
Recall	Knowledge	Retention	+3	+3	12/21	B
Replit Agent	Code	Delegation	+2	+2	11/21	B-
ChatGPT	Chat	Delegation	-2	+2	8/21	C+
Notion AI	Knowledge	Delegation	-2	+2	7/21	C+
CapCut	Video	Delegation	0	+1	6/21	C

Sovereignty scored on a 21-point scale across seven dimensions of user agency (Agency Over Input, Process Transparency, Output Verifiability, Control During Execution, Verification Support, Outcome Ownership, Override Capability — each 0 to 3). This is a separate assessment from the OL component scores. Full component scores, temporal estimates, and boundary assessments available in the practitioner appendix. All scores from a single-assessor methodology; inter-rater reliability not yet established.

V. Eight principles for the conductor

These are design principles distilled from the framework and consistent across all 10 audits. Each is stated as a principle, then translated into what it means for daily UX practice.

1. Articulation before amplification The user states their position, criteria, or intent before the AI contributes. This single pattern is the strongest differentiator between effective and wasteful AI interaction. In practice: design input flows that ask "what are you looking for?" before showing AI suggestions. Never lead with the AI's answer.

2. Preserve productive friction Reduce coordination overhead, but keep verification effort. The goal is not a frictionless experience — it's an experience where the friction falls in the right places. In practice: make it easy to *see* what the AI did. Don't make it easy to *skip* evaluating what the AI did.

3. Scaffold, don't replace Guide users toward independent capability, then fade the scaffolding. AI assistance should be a training wheel, not a permanent crutch. In practice: track whether users become more capable over time, not just more productive. If usage increases but capability doesn't, the tool is creating dependency.

4. Schema correction over skill addition Most AI tool failure traces to users applying the wrong mental model (search-engine thinking applied to AI), not lacking specific skills. In practice: onboarding should reframe how the tool works, not just teach what buttons to press. The most effective intervention in a study of over 12,000 users isn't prompt training — it's helping users understand that AI isn't search.

5. Strategic friction is a feature Deliberate pause points at decision moments preserve independent judgment. In practice: before a user accepts AI-generated content into their final output, insert a moment of conscious decision. Not a confirmation dialogue — a design moment that makes the choice visible.

6. Compound, don't transact Each interaction should make the next one better. In practice: design for session continuity. What did the user learn from this interaction that carries forward? If every session starts from zero, the tool is a slot machine regardless of how good the AI is.

7. Temporal vigilance over session trust Output quality at Turn 1 does not predict quality at Turn 10. In practice: test your AI features over sustained sessions, not single exchanges. Build drift detection into the interaction — subtle reminders of original constraints, periodic quality re-anchoring, session segmentation for long tasks.

8. Boundary preservation over workflow speed Moving work between tools quickly is not the same as moving it well. In practice: at tool transitions, help users carry over their reasoning and quality standards, not just the output file. The fastest export is worthless if it brings degraded standards into the next context.

VI. What this means for your work tomorrow

The design element test

Before shipping any AI-powered feature, run it through four questions:

1. Does it reduce Coordination Cost (Cc) or Context Maintenance (Cm)? → It reduces unproductive overhead. Good.
2. Does it preserve Verification Capacity (Cv) or Cognitive Reserve (Cr)? → It keeps the user thinking. Essential.
3. Does it resist Temporal Degradation (Ct)? → It works at Turn 10, not just Turn 1. Test it.
4. Does it minimise Cross-boundary Load (Cx)? → It plays well with other tools. Check it.

If a design element fails all four, it doesn't belong in the product. If it passes only the first and fails the second, you've built a frictionless path to worse outcomes.

Two higher-order tests complete the evaluation:

The Sovereignty Test: Does this element develop the user's capacity to think independently, or does it create dependency on the tool?

The Persistence Test: If this tool disappeared tomorrow, what would the user have gained that persists?

The conductor's constraints

Six tactical interventions for managing temporal and boundary effects in AI features — practical enough to put in a design review checklist:

Quality anchoring — Establish explicit quality criteria at session start. Surface them visibly. Make them available throughout, not just at the beginning.

Constraint persistence — Re-present critical parameters periodically. Instructions given at Turn 1 decay by Turn 5. Design for repetition, not one-time input.

Session segmentation — Break sustained AI interactions into bounded segments with natural reset points. Long unbroken sessions are where degradation accumulates.

Cognitive air-gapping — Design deliberate context resets that interrupt self-referential baseline formation. This might feel disruptive. It's protective.

Calibration re-anchoring — Periodically surface comparison between current output and original standards. Don't rely on users to remember what "good" looked like eight turns ago.

Boundary bridging — At tool transitions, explicitly carry over reasoning chains, constraints, and quality criteria — not just the output file. Design the handoff, not just the export.

Where Orchestration Load sits in the organisation

Orchestration Load Management is not a rebrand of UX. It's an expansion. The skills are adjacent: user research, cognitive modelling, interaction design, longitudinal testing. The problems are new: temporal degradation, verification design, sovereignty preservation, cross-boundary coherence.

In practice, this means UX teams working on AI products need to add three capabilities they probably don't have today: longitudinal session testing (not just single-task usability), verification design as a first-class pattern library, and cross-tool workflow assessment.

It also means the metrics change. Task completion time and user satisfaction — the traditional UX North Stars — become insufficient. A user who completes tasks quickly and reports high satisfaction while their cognitive capability degrades is not a success story. It's a slow-moving failure that current measurement can't see.

New metrics for the conductor's toolkit: verification accuracy over time, skill persistence after tool removal, sovereignty score across extended sessions, and output quality at Turn 10 versus Turn 1.

VII. The seat you already have

There is a window right now, and it's not going to stay open long.

AI product teams need someone who understands cognitive load, designs for human capability, and thinks in terms of user journeys rather than feature specs. They need someone who can look at a "Thought for 12 seconds" progress bar and recognise that the loading-bar pattern borrowed from deterministic tools is actively harmful in a probabilistic system. They need someone who can translate between what the model can do and what the human needs to remain capable of doing.

That description is a UX practitioner with an expanded toolkit.

The alternative is that this territory defaults to engineering ("we'll add a confidence score") or product management ("the satisfaction metrics look fine"). Neither discipline is trained to see the cognitive relationship between the human and the system. Neither has the methodology to measure it over time. Neither will prioritise it — because the immediate metrics look good, and the damage is longitudinal.

The Orchestration Load Framework is not a competing discipline. It's the next chapter of yours. The same rigour that built modern UX practice — the insistence on understanding the human, measuring what matters, and designing for real outcomes rather than surface metrics — is exactly what AI interaction needs now.

The craft doesn't change. The scope does. And the practitioners who claim this territory first will define the field for a generation.

Appendix A: The complete OL scoring system

This appendix provides the full measurement methodology for practitioners ready to audit AI tools or evaluate AI features. It is intentionally detailed — the main paper gave you the "why" and "what." This gives you the "how."

Surface decomposition

Every AI tool decomposes into seven logical interaction surfaces. Each surface is scored independently before aggregation:

Surface	Function
Entry	Where the user provides input
Output	Where the AI presents results
Context	Where prior state is visible
Control	Where the user directs AI behaviour
Navigation	Where the user moves through the tool
Feedback	Where the user corrects or refines
Knowledge	Where accumulated understanding lives

Component scoring rubrics

Cc (Coordination Cost) — **Goal: Minimise**

Score	Meaning
0	No coordination support
-1 to -2	Minimal — basic prompt input
-3 to -4	Moderate — smart defaults, templates, auto-context
-5 to -6	Strong — AI manages workflow, surfaces context
-7	Maximum — near-invisible coordination

Cv (Verification Capacity) — Goal: Preserve

Score	Meaning
-2	Net negative — false confidence signals (Cosmetic Metacognitive Narration)
0	Neutral — no support, no undermining
+1 to +2	Minimal — basic display, some comparison
+3 to +4	Moderate — citations, diffs, confidence indicators
+5 to +6	Strong — inline verification, traceable evidence chains

Cm (Context Maintenance) — Goal: Minimise cost

Score	Meaning
0	No support — every session starts from zero
-1 to -2	Minimal — history exists but not integrated
-3 to -4	Moderate — workspace persistence, cross-session continuity
-5 to -6	Maximum — persistent spatial structures, automatic retrieval

Cr (Cognitive Reserve) — Goal: Preserve

Score	Meaning
0	No preservation — user manages tool, not thinking
+1	Minimal — some automation
+2 to +3	Moderate — structural offloading frees capacity
+4 to +5	Maximum — spatial externalisation, visible pipeline

Ct (Temporal Degradation) — Goal: Minimise

Score	Meaning
+2	Temporally Protective — re-anchoring mechanisms present
+1	Temporally Stable — no measurable drift
0	Neutral — adequate for short sessions

Score	Meaning
-1	Degrading — measurable output drift
-2	Corrosive — drift combined with user calibration shift
-3	Dangerous — self-referential degradation loop confirmed

Cx (Cross-boundary Load) — Goal: Minimise

Score	Meaning
+2	Boundary Bridging — active transition support
+1	Boundary Aware — clean export, minimal friction
0	Neutral — standard tool transition
-1	Friction — measurable reconstruction cost
-2	Corrosive — calibration contamination, cognitive gravity
-3	Destructive — workflow worse than either tool alone

Design pattern assessment

Six empirically-grounded patterns scored on a 4-level scale:

Pattern	Purpose
Metacognitive Narration	AI shows its reasoning process
Articulation Before Amplification	User states position first
Risk Bands	Uncertainty made visible
Sovereignty Scaffolds	Structural supports for user agency
Epistemic Pause	Deliberate friction at decision points
Feedback Loop Architecture	Correction and refinement mechanisms

The 4-Level Scale:

Level	Label	Impact
0	Absent	Neutral

Level	Label	Impact
1	Cosmetic	Worse than Absent — creates false confidence
2	Functional	Partial benefit — works but inconsistent
3	Integrated	Full benefit — deeply woven, consistent

Critical finding across all 10 audits: Level 1 (Cosmetic) scores worse than Level 0 (Absent). A loading bar that says "Thinking..." creates more false confidence than no indicator at all.

The sovereignty assessment

Seven core dimensions form the base assessment, scored 0–3 each (21 points total). Two additional dimensions (Temporal Stability, Boundary Preservation) extend the assessment for tools evaluated under the full temporal and boundary audit protocols, bringing the maximum to 27.

Core Dimensions (used in the 10-tool audit):

#	Dimension	Max
1	Agency Over Input	3
2	Process Transparency	3
3	Output Verifiability	3
4	Control During Execution	3
5	Verification Support	3
6	Outcome Ownership	3
7	Override Capability	3

Extended Dimensions (for temporal/boundary audits):

#	Dimension	Max
8	Temporal Stability	3
9	Boundary Preservation	3

Sovereignty score was the single strongest predictor of overall grade in the initial assessment dataset (using the 7 core dimensions, max 21):

Sovereignty (out of 21)	Predicted Grade
≤ 8	C-tier
9–11	C+ to B-
12–16	B to B+
17–19	B+ to A-
20–21	A

Composite grading

Grade	Requirements
A	Sovereignty ≥ 18/21, Skill Dev ≥ 2/3, ≥4/6 patterns at Level 2+, no critical findings
B+	Sovereignty ≥ 14/21, Cv ≥ +4, Cr ≥ +4, comprehensive patterns
B	Sovereignty ≥ 12/21, Cv ≥ +3, most patterns at Level 2+
B-	Sovereignty ≥ 10/21, Cv ≥ +2, some pattern coverage
C+	Sovereignty 7–9/21, Cv ≤ +1, minimal patterns
C	Sovereignty ≤ 6/21, zero effective pattern implementation

Critical findings (e.g., Cosmetic Metacognitive Narration creating false reasoning confidence; sycophancy reinforcing bias; dependency-inducing frictionless completion) cap grade at C+ regardless of other scores. Temporal degradation ≤ -2 or cross-boundary load ≤ -2 each lower the grade by one full step.

Full 10-tool audit results

Tool	Domain	Category	Cc	Cm	Cv	Cr	Est. Ct	Est. Cx	Sov.	Grade
UNPRODUCTIVE					PRODUCTIVE		TEMPORAL		AGENCY	
			<i>Minimise ↓</i>		<i>Preserve ↑</i>		<i>Minimise ↓</i>			
ChatGPT	Chat	Delegation	-5	-3	-2	+2	-2	-1	8/21	C+
Replit Agent	Code	Delegation	-4	-3	+2	+2	-1	-1	11/21	B-
CapCut	Video	Delegation	-5	-2	0	+1	0	-1	6/21	C

Tool	Domain	Category	Cc	Cm	Cv	Cr	Est. Ct	Est. Cx	Sov.	Grade
UNPRODUCTIVE				PRODUCTIVE		TEMPORAL		AGENCY		
			Minimise ↓		Preserve ↑		Minimise ↓			
Descript	Video	Synthesis	-4	-4	+4	+3	0	0	14/21	B
Notion AI	Knowledge	Delegation	-7	-5	-2	+2	-1	-2	7/21	C+
Notebook LM	Knowledge	Synthesis	-5	-4	+6	+4	+1	-1	16/21	B+
Recall	Knowledge	Retention	-4	-5	+3	+3	0	+1	12/21	B
Cursor	Code	Synthesis	-5	-4	+6	+3	0	-1	16/21	B
Heptabase	Spatial	Externalisation	-4	-6	+4	+5	+1	0	16/21	B+
Krea Nodes	Spatial	Externalisation	-3	-3	+4	+4	0	-1	15/21	B

All scores from single-assessor methodology. Ct and Cx values are architectural estimates based on tool analysis, not empirical measurements from temporal or boundary audit protocols. Inter-rater reliability has not been established. These results should be interpreted as a consistent initial assessment inviting independent replication, not as validated psychometric findings.

The audit process

Four-phase methodology for practitioners:

Phase 1 — Surface decomposition. Identify the seven interaction surfaces. Map them to screenshots. Flag missing surfaces (absence is diagnostic — a missing Knowledge surface means no persistence).

Phase 2 — Per-surface scoring. Score four OL components + six design patterns per surface. Use the rubrics above. Document evidence for each score.

Phase 3 — Cross-surface analysis. Flow analysis across surfaces. Scaffolding assessment. Sovereignty evaluation across the nine dimensions.

Phase 4 — Synthesis and grading. Aggregate scores. Identify critical findings. Assign composite grade.

Optional phase 5 for extended assessment: Temporal Audit (10-turn protocol adding Ct scores) and Boundary Audit (3-configuration comparison adding Cx scores).

Appendix B: Key research

Researcher	Contribution
Ethan Mollick (Wharton)	Field experiments on AI-augmented knowledge work; centaur/cyborg taxonomy
Fabrizio Dell'Acqua (HBS)	The jagged technological frontier in AI assistance
Mark Steyvers (UC Irvine)	AI calibration and human-AI decision making
Ma et al. (CMU)	Four-Dimensions AI competency framework
Bastani et al.	Randomised controlled trial on AI scaffolding effects (PNAS)
Shen & Tamkin	AI skill formation impacts ($d = 0.738$ skill deficit from unscaffolded use)
Cheng et al.	Mental model elicitation study (N=12,000+)
Wei Xu	Human-AI Joint Cognitive Systems
John Sweller	Cognitive Load Theory
Christopher Wickens	Multiple Resource Theory

Appendix C: Framework maturity and known limitations

What has been established: The six-component OL model consistently differentiates tool quality across 10 independent audits in 6 domains. The sovereignty assessment reliably predicts composite grade. The design pattern scale discriminates between cosmetic and functional implementation. Paradigm outpredicts features as a determinant of tool quality.

What has not been established: OL as a formal psychometric construct (no scale development or neuroimaging validation). Inter-rater reliability (single assessor throughout). User outcome data (methodology assesses architecture, not actual user performance). Ct and Cx as empirical measurements (protocol defined, retrospective estimates only). External replication of any component.

What this means: The framework is a well-evidenced working model, not a validated instrument. It produces consistent, discriminating results and is grounded in established cognitive science. It is not yet a peer-reviewed measurement system. Use it as a powerful diagnostic lens and a rigorous design evaluation tool — and maintain appropriate epistemic humility about its maturity.

"The tool that makes thinking easier is not the same as the tool that makes thinking better. The question every AI interaction must answer: which kind are you building?"

ORCHESTRATION_LOAD_FRAMEWORK_WHITEPAPER_v2_0.md Date: 2026-02-24 Full reference: Parts 1-5 of the Orchestration Load Framework (v1.1-v1.5)