LIFELONGCUAWORLD: BENCHMARKING COMPUTER-USE AGENTS FOR LIFELONG LEARNING IN HYBRID ENVIRONMENTS WITH CLI KNOWLEDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Computer-use agents have advanced in task execution, but reliable lifelong assistance remains underexplored. We introduce **LifelongCUAWorld** to evaluate the capability profile of *lifelong computer-use agents* through 180 expert-level tasks across 19 applications, with GUI/CLI/code interfaces and a structured CLI knowledge base. *Expert* evaluates independent tasks through verifiable deliverables; *Profile* tests acquisition and reuse of hidden preferences across task sequences for the same user; *Revision* provides staged feedback on agents' drafts and evaluates feedback requirements and preservation of previously correct content. Ten evaluated systems show partial professional competence, but none strictly completes an Expert task. DeepSeek Harness raises DeepSeek-V4.1-Flash's Expert score from 0.098 to 0.239. Six systems never request clarification, and preference-related delivery compliance reaches at most 6.3%. Agents can complete feedback rounds and preserve correct content, yet feedback compliance reaches at most 17.2%. Ablations support the CLI knowledge base and benchmark design: task-relevant CLI knowledge improves Expert scores, supplied prior user information improves Profile compliance, and Revision ablations support assessing feedback compliance and content preservation separately. Current systems still fall short of reliable lifelong assistance; LifelongCUAWorld provides a systematic evaluation platform for advancing *lifelong computer-use agents*. Anonymous Project

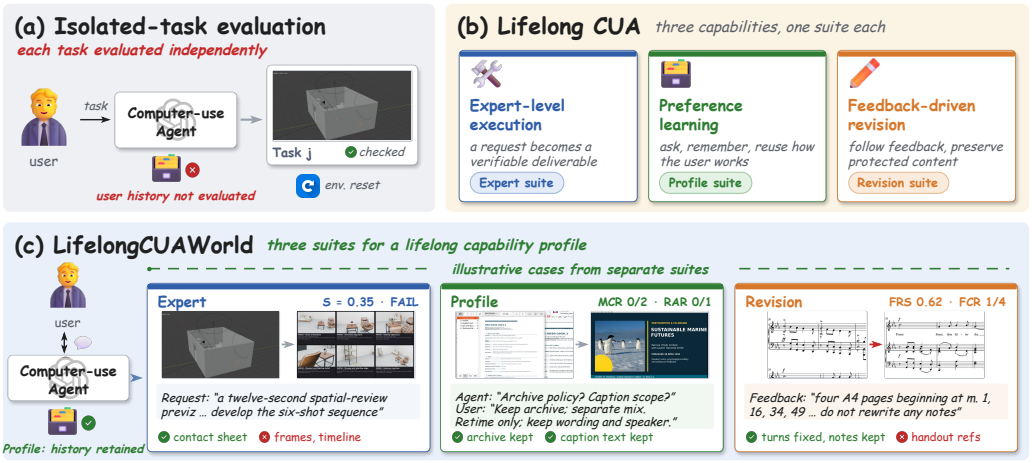


Figure 1: From one-shot execution to lifelong assistance. A professional user entrusts work to an agent, expects their preferences to guide later tasks, and returns with feedback to refine the deliverable. Existing benchmarks assess parts of this collaboration; LifelongCUAWorld brings professional execution, cross-task preference learning, and feedback-driven revision into one capability profile.

1 INTRODUCTION

Computer-use agents (CUAs) based on large language models (LLMs) have made progress in executing complex tasks in real software (OpenAI, 2024; Zheng et al., 2024; Zhang et al., 2025; Tang et al., 2025). On OSWorld, state-of-the-art success rates rose from about 12% to over 80% within

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

a year (Xie et al., 2024). Yet learning from repeated interactions with the same user remains underexplored in existing benchmarks (Koh et al., 2024; Rawles et al., 2025). We envision *lifelong computer-use agents* as personalized assistants that understand users' working habits, collaborate on professional tasks, and continually learn and improve.

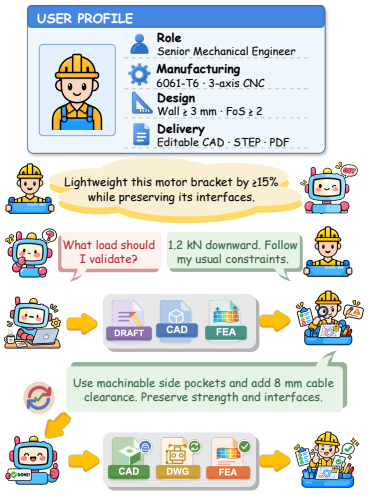


Figure 2: A lifelong CUA at work. The agent clarifies the load, follows the engineer's usual constraints, and revises the motor bracket based on feedback.

In practice, users therefore expect agents to handle professional work, understand their habits and preferences, and iteratively refine deliverables with them through feedback to meet delivery requirements (Figures 1 and 2). This requires **expert-level work**, coordinating software tools to produce usable, verifiable results (Sun et al., 2026; Yuan et al., 2026; Li et al., 2026). Agents must learn **user preferences across tasks**, using history and dialogue to inform later work and asking for clarification when needed (Jang et al., 2026; Chen et al., 2026). **Feedback-driven revision** requires implementing requested changes while preserving correct content. These requirements define a *capability profile* for *lifelong computer-use agents*.

> RESEARCH QUESTION How can we systematically evaluate whether a lifelong computer-use agent completes expert-level work, accumulates and applies user knowledge across tasks, and refines deliverables as requirements evolve?

Existing benchmarks cover parts of this profile. Agents' Last Exam evaluates verifiable professional tasks (Sun et al., 2026). Others examine multi-application workflows (Yuan et al., 2026; Li et al., 2026), personalized single tasks (Jang et al., 2026; Kong et al., 2026; Chen et al., 2026), or user input in one episode (Wang et al., 2026). Evaluating lifelong assistance requires task sequences to test user knowledge reuse and successive submissions to assess feedback compliance and content preservation.

We introduce LifelongCUAWorld to evaluate this profile in hybrid GUI/CLI/code desktop environments, supported by a structured CLI knowledge base derived from professional software. LifelongCUAWorld-Expert comprises 72 independent tasks across 13 domains and 19 applications; 92% involve multiple applications. LifelongCUAWorld-Profile provides nine-task sequences for six professional users (54 tasks), with preferences accessible only through history and `ask_user`. LifelongCUAWorld-Revision provides 54 multi-stage episodes, predefined feedback after intermediate submissions, and annotations for feedback requirements and protected content.

Experiments on LifelongCUAWorld yield three observations. (1) **Agents show partial professional competence but struggle with complete delivery.** With DeepSeek-V4.1-Flash, DeepSeek Harness raises the Expert score from 0.098 to 0.239, yet no evaluated system strictly completes an Expert task. (2) **Agents rarely seek missing user information.** Six of ten systems never invoke `ask_user`. Across systems, 91–100% of required clarification slots remain unresolved, and preference-related delivery compliance reaches at most 0.063. (3) **Agents struggle to satisfy user feedback in their deliverables.** Even the system with the highest feedback compliance satisfies only 17.2% of evaluated feedback requirements. Ablations support the CLI knowledge base and evaluation design. Task-relevant CLI knowledge improves Expert scores, and supplying prior oracle Q/A improves performance on later Profile tasks. Resetting the Revision workspace reduces CPR while FCR changes little, supporting their complementary diagnostic roles. Our contributions are threefold:

- **We introduce LifelongCUAWorld, a benchmark for evaluating lifelong computer-use agents.** We characterize these agents through a three-part capability profile: expert-level execution, cross-task preference learning, and feedback-driven revision. The benchmark contains 180 expert-level desktop tasks in hybrid GUI/CLI/code environments, comprising LifelongCUAWorld-Expert (72 tasks) and LifelongCUAWorld-UserCollab (108 tasks across Profile and Revision).
- **We develop reproducible protocols and metrics for evaluating user collaboration.** The protocols test whether agents elicit and reuse user preferences across tasks and revise their own deliverables through successive rounds of feedback. The metrics report preference-related delivery, feedback satisfaction, and content preservation alongside overall task performance.

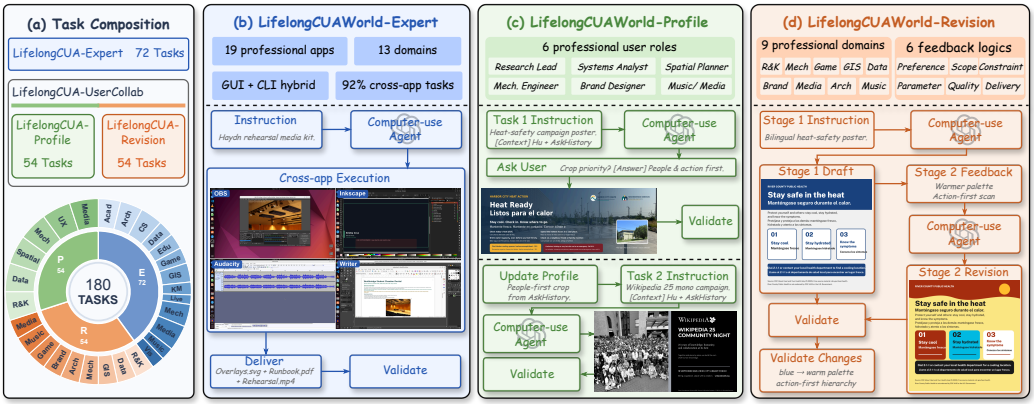


Figure 3: Overview of LifelongCUAWorld. The three suites evaluate professional task execution and sustained user collaboration. LifelongCUAWorld-Expert delivers verifiable artifacts through cross-application execution, while LifelongCUAWorld-Profile reuses preferences elicited through `ask_user` across tasks. LifelongCUAWorld-Revision releases feedback after submission and evaluates revisions for feedback compliance and preservation of previously correct content.

- **We provide an extensible, capability-driven scenario modeling framework and evaluate ten agent systems.** It builds tasks from professional roles and workflows and provides a structured CLI knowledge base from professional software. Ablations test how CLI knowledge, supplied user information, and draft source affect performance across suites.

2 LIFELONGCUAWORLD

2.1 PROBLEM FORMULATION

A lifelong computer-use agent is defined by three capabilities (Figure 1b): *expert-level execution*, turning a high-level goal into a usable and verifiable deliverable; *cross-task preference learning*, acquiring a user's preferences and applying them in later tasks; and *feedback-driven revision*, improving an existing deliverable as the user's requirements evolve. Expert-level execution underpins all 180 tasks in LifelongCUAWorld. LifelongCUAWorld-Expert evaluates this capability in isolation through 72 independent tasks. LifelongCUAWorld-UserCollab (108 tasks) builds on this foundation with LifelongCUAWorld-Profile and LifelongCUAWorld-Revision, which additionally assess cross-task preference learning and feedback-driven revision, respectively (54 tasks each; Figure 3a). Table 1 summarizes their composition across 19 professional applications.

Table 1: Composition and protocols of the three suites. Apps: distinct applications (mean per task); Steps: estimated range, budget = 2×.

	Expert	Profile	Revision
Tasks	72	54	54
Structure	13 domains	6 roles × 9 tasks	9 scenarios × 1–3 rounds
Apps	19 (2.7)	16 (2.8)	15 (3.2)
Steps	38–130	56–156	88–236
User	one instruction	H_u, $A_{a,u}^{<j}$, `ask_user`	request, staged feedback
State kept	none	H_u, $A_{a,u}$	workspace (episode)

In Expert, the user sets the goal and leaves execution to the agent. In Profile, the user remains available for clarification throughout execution; in Revision, the user returns with feedback to refine the deliverable. Each benchmark instance combines a base task $\tau = (g, s_0, X, \mathcal{D}, \mathcal{E})$ with one of these interaction protocols. The task specifies a high-level goal g, an initial desktop state s_0, input assets X, required deliverables $\mathcal{D}$, and evaluator units $\mathcal{E}$. The protocol fixes what the agent sees of the user, how it may interact, and which state survives between episodes (Table 1, Algorithm 1). Predefined answers and feedback reduce response variability, supporting reproducible comparisons.

Professional scenario modeling. We design tasks for all three suites using professional materials collected online. These include papers, datasets, technical documentation, and open-source projects, primarily published or updated within the preceding two years. We define professional

Algorithm 1 Profile protocol (left; agent-initiated clarification) and Revision protocol (right; feedback released per submission)

Input: user u with hidden P_u, history H_u, tasks $\tau_{u,1..9}$
$A \leftarrow \emptyset$
for $j = 1, \ldots, 9$ **do**
  reset to $s^0_{u,j}$; run agent on $(g_{u,j}, X_{u,j}, H_u, A)$
  answer `ask_user` using P_u during execution
  append the valid exchanges of task j to A
  score $\hat{D}_{u,j}$ on task and profile checkpoints
end for

Input: request $g^{(0)}$, feedback $f^{(1..K)}$, units $\mathcal{R}^{(k)}$, $\mathcal{P}^{(k)}$
$k \leftarrow 0$; run agent on $g^{(0)}$ until it submits
while $k < K$ **do**
  snapshot $D^{(k)}$; $k \leftarrow k + 1$; release $f^{(k)}$
  run to submission on same workspace with $g^{(0)}$, $f^{(1..k)}$
end while
snapshot $D^{(K)}$; replay snapshots offline
score $\mathcal{R}^{(k)}$, $\mathcal{P}^{(k)}$ for every reached stage (Eq. 2)

roles, workflows, and deliverables from these resources, selecting scenarios for *professional relevance*, *workflow closure*, and *artifact verifiability*. Application handbooks summarize software capabilities; CLI-Anything (Yang et al., 2026) extracts command-line capabilities for a structured CLI knowledge base. Profile adds user history and preference-dependent requirements; Revision adds staged feedback and preservation requirements (Appendix D).

2.2 LIFELONGCUAWORLD-EXPERT: EXPERT-LEVEL TASK COMPLETION

In LifelongCUAWorld-Expert, the user provides complete requirements upfront, and the agent must produce a usable, verifiable professional deliverable. The 72 tasks follow this design as independent episodes, each with an instruction, an initialized workspace, and input assets. Tasks call for domain-specific interpretation, composition of several applications, and maintenance of project state over an estimated 38–130 steps. Required intermediate artifacts allow partial completion to be evaluated.

2.3 LIFELONGCUAWORLD-PROFILE: CROSS-TASK USER PREFERENCE LEARNING

LifelongCUAWorld-Profile adds the user as a persistent presence. We derive six user archetypes from reference tasks in existing computer-use benchmarks (Appendix E.1). Stable work habits, factual attributes, and behavioral preferences form each user's *hidden* structured profile P_u, including software-use conventions, preferred deliverable formats, and quality expectations. The agent learns about u through a natural-language history H_u of prior requests, revisions, and software habits, and through its own questions. All agents serving u receive the same H_u; P_u stays hidden throughout.

For each user, a nine-task sequence assesses preference *discovery*, *reuse* with selective clarification, and *adaptation* to changed conditions. Let $A^{<j}_{a,u}$ denote the `ask_user` exchanges recorded for agent a in the first $j-1$ tasks of user u. At task j, the agent receives $(g_{u,j}, s^0_{u,j}, X_{u,j})$, H_u, $A^{<j}_{a,u}$, and the `ask_user` interface. Valid exchanges are appended to form $A^{<j+1}_{a,u}$. The workspace is reset between tasks while H_u and $A^{<j}_{a,u}$ persist. Agents serving the same user share an initial context and develop user models from their own interaction histories. `ask_user` queries are answered by a deterministic slot-based responder with one canonical answer per predefined topic (Appendix E.3).

2.4 LIFELONGCUAWORLD-REVISION: ITERATIVE DELIVERABLE REVISION

LifelongCUAWorld-Revision evaluates iterative refinement of professional deliverables in 54 episodes across nine scenarios. Each non-final submission saves a snapshot and releases pre-authored feedback based on expected stage artifacts. Feedback is fixed across agents, not regenerated from their submissions. The workspace persists until submission after the final round. Artifact validity is assessed offline. Each episode follows the trajectory $g^{(0)} \to D^{(0)} \xrightarrow{f^{(1)}} D^{(1)} \xrightarrow{f^{(2)}} \cdots \xrightarrow{f^{(K)}} D^{(K)}$, where $g^{(0)}$ denotes the initial request, $D^{(0)}$ the first draft, and $f^{(k)}$ the user's feedback at stage k. *Revision units* $\mathcal{R}^{(k)}$ specify stage feedback requirements, including explicit constraints to keep content unchanged; *preservation units* $\mathcal{P}^{(k)}$ specify designated protected requirements. The sets can concern overlapping properties. We score feedback satisfaction after revision and retention of previously valid protected content (Appendix F).

2.5 DELIVERABLE-CENTERED EVALUATION

All three suites use a shared evaluation framework to assess the validity and professional quality of the submitted deliverables $\hat{\mathcal{D}}_\tau$ (Figure 4; Appendix G). The **rule layer** uses deterministic check-

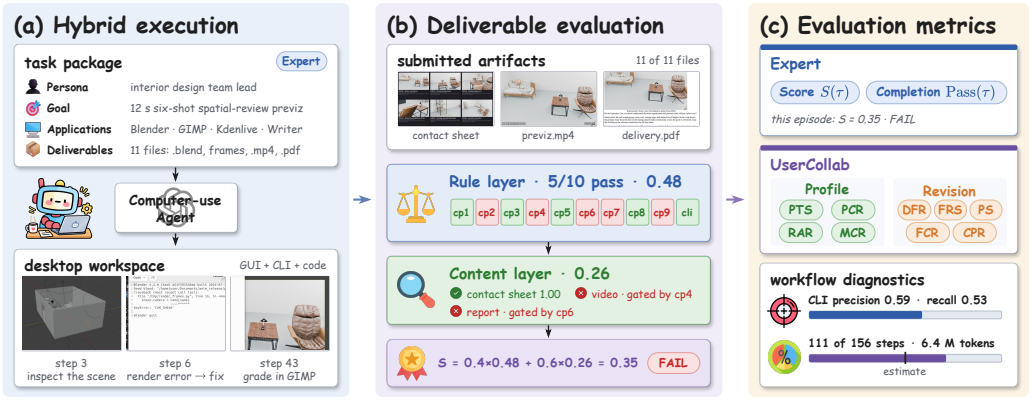


Figure 4: Execution and evaluation workflow. (a) The agent executes an Expert task through code, CLI, and GUI actions. (b) Submitted artifacts undergo rule checks that gate rubric-based content evaluation. (c) Suite-specific metrics report task performance; execution diagnostics record CLI use, steps, and token consumption.

points to inspect file presence and format, required content, and native application state. Their weighted mean defines $S_{\text{rule}} \in [0, 1]$. The **content layer** combines deterministic content checks with rubric-based judgments of composition, semantic consistency, and professional completeness, using evidence extracted from the artifacts. Validity checks gate content credit: a failed prerequisite sets the dependent scores to zero. The weighted mean after gating defines $S_{\text{content}} \in [0, 1]$. We report a task score and a strict pass indicator:

$$S(\tau) = G_\tau \cdot \mathrm{Agg}_\tau(S_{\text{rule}}, S_{\text{content}}), \qquad \mathrm{Pass}(\tau) = \mathbb{I}[S(\tau) \geq \theta_\tau \ \wedge \ \text{all critical checks pass}], \quad (1)$$

Here, $G_\tau \in [0, 1]$ is a run-level completion and validity factor. Each task specifies its aggregation function Agg_τ, passing threshold θ_τ, and critical checks before evaluation. The continuous score credits partial progress, while the binary indicator records strict task completion.

Suite-specific metrics. Expert reports deliverable quality $S(\tau)$ and strict completion $\mathrm{Pass}(\tau)$. Profile Task Score (PTS) measures performance; Preference Compliance Rate (PCR) scores P_u-based checkpoints, including delivery and evidence conditions. Redundant Ask Rate (RAR) and Missed Clarification Rate (MCR) measure redundant calls and unresolved required slots (Appendix E.4). Revision reports initial-stage Draft Failure Rate (DFR) and Final Revision Score (FRS), the mean final deliverable score. We pool eligible units $\mathcal{R}$ and $\mathcal{P}$ over reached feedback rounds:

$$\mathrm{FCR} = \frac{|\mathcal{R}^+|}{|\mathcal{R}|}, \qquad \mathrm{PS} = \frac{|\mathcal{P}^+|}{|\mathcal{P}|}, \qquad \mathrm{CPR} = \frac{|\mathcal{P}^{\circ +}|}{|\mathcal{P}^\circ|}, \quad (2)$$

$^+$ denotes satisfaction after revision; $\mathcal{P}^\circ \subseteq \mathcal{P}$ contains units valid in the prior snapshot. FCR scores feedback satisfaction, PS protected requirements, and CPR retention of previously correct content. Interpret these rates with stage and unit coverage. Strict completion requires passing all stage gates and critical checks (Appendix F.2).

3 Experiments

3.1 Experimental Setup

Evaluation subsets. Main experiments use fixed 36-task subsets for each suite. Expert retains approximately half the tasks in each of the 13 domains, covering all 19 applications. Profile selects six tasks per role track in their original order. The selection preserves cross-task dependencies: preferences needed by a reuse task can be elicited in an earlier selected task for the same user. Revision includes four episodes from each of the nine scenarios, with one to three feedback rounds. Subset compositions are detailed in Appendices C.3–C.6.

Evaluated systems and configuration. We evaluate an agent harness and two groups of model baselines (Table 2): **(a) Agent Harness:** DeepSeek Harness (DeepSeek-AI, 2026a) paired with DeepSeek-V4.1-Flash (DeepSeek-AI, 2026b); **(b) Proprietary models:** GPT-5.4, GPT-5.6 Sol, GPT-5.6 Luna, Seed-2.1-Pro, and Qwen3.7-Plus (Qwen Team, 2026); and **(c) Open-weight models:**

Table 2: Main results on the 36-task subsets. Model baselines can use GUI, CLI, and code. All systems, including DeepSeek Harness, receive CLI knowledge. FCR/PS/CPR are conditional rates; coverage is reported in Table 17. RAR is undefined without queries, CPR without previously valid preservation units. Cost: mean USD/task; **best** and second-best highlighted.

	LifelongCUAWorld-Expert			LifelongCUAWorld-Profile					LifelongCUAWorld-Revision					
System	**Score↑**	**CLI F1↑**	**Cost↓**	**PTS↑**	**PCR↑**	**RAR↓**	**MCR↓**	**Cost↓**	**DFR↓**	**FRS↑**	**FCR↑**	**PS↑**	**CPR↑**	**Cost↓**
Agent Harness														
DeepSeek Harness														
+ DeepSeek-V4.1-Flash	**0.239**	0.235	4.55	**0.218**	**0.063**	**0.00**	0.93	4.89	**0.943**	**0.116**	**0.172**	0.433	0.956	11.42
Proprietary models														
GPT-5.4	0.116	0.124	20.36	0.044	0.046	0.50	0.93	10.04	0.958	0.022	0.097	0.400	0.667	32.67
GPT-5.6 Sol	0.097	0.283	20.52	0.062	0.028	**0.00**	0.98	20.12	0.972	0.046	0.118	**0.439**	0.955	50.98
GPT-5.6 Luna	0.061	0.220	1.65	0.064	0.028	–	1.00	1.73	1.000	0.016	0.069	0.311	0.786	3.52
Seed-2.1-Pro	0.067	0.291	7.66	0.103	0.049	–	1.00	12.53	0.971	0.022	0.016	0.246	0.500	18.65
Qwen3.7-Plus	0.058	**0.519**	1.99	0.047	0.009	–	1.00	1.63	1.000	0.021	0.047	0.215	**0.957**	3.02
Open-weight models														
DeepSeek-V4.1-Flash	0.098	0.246	2.27	0.107	0.032	**0.00**	**0.91**	2.96	0.972	0.068	0.118	0.339	0.941	6.88
GLM5.3-flash	0.114	0.377	**0.48**	0.104	0.049	–	1.00	**0.42**	0.971	0.066	0.065	0.282	0.914	**1.02**
Kimi K3	0.028	0.059	3.09	0.051	0.042	–	1.00	4.08	1.000	0.000	0.000	0.111	–	1.97
MiniMax M3	0.030	0.243	3.57	0.054	0.035	–	1.00	5.42	1.000	0.008	0.011	0.022	0.333	6.56

DeepSeek-V4.1-Flash, GLM5.3-flash (GLM-5-Team, 2026), Kimi K3 (Kimi Team, 2026), and MiniMax M3 (Lai et al., 2026). DeepSeek Harness receives our CLI knowledge base. Model baselines in (b) and (c) have access to GUI interaction, CLI commands, and code execution, with task-relevant application registries provided as CLI knowledge. All systems use a per-task step limit of twice the estimated step count. All systems use the same evaluation subsets and suite protocols. In Profile, P_u remains hidden; agents receive H_u, their own prior exchanges $A_{a,u}^{<j}$, and `ask_user`. Model-specific decoding and context settings are given in Appendix I.1. Execution statistics are reported in Table 3, and average costs in Appendix C.4.

Evaluation protocol. All three suites use a shared artifact-based evaluation framework, with suite-specific metrics for their target capabilities. The evaluator combines task-specific rule verifiers with audited rubrics, using Gemini-3.1-Pro as the content judge. Each content checkpoint receives three judgments aggregated with a majority veto (Appendices G.1 and H.3).

3.2 Expert-Level Task Completion

Current models show partial competence in professional task execution but fall short of complete delivery. All evaluated systems earn partial credit, but none strictly completes an Expert task (Table 2). The model baselines score 0.028–0.116, reflecting partial fulfillment of professional requirements. A suitable agent harness can further improve professional task execution. With the same DeepSeek-V4.1-Flash model, DeepSeek Harness raises the score from 0.098 to 0.239 and earns credit on 27 of 36 tasks. This improvement comes with higher token consumption, averaging 14.84M tokens per task versus 6.53M for the model baseline (Appendix C.4). GLM5.3-flash approaches GPT-5.4's score (0.114 vs. 0.116) with fewer steps per task (67.6 vs. 107.7; Table 3) and the lowest estimated API cost ($0.48 per task).

3.3 Proactive User Preference Learning

Agents rarely seek missing user information. Six of the ten systems never invoke `ask_user`. GPT-5.4, DeepSeek-V4.1-Flash, DeepSeek Harness, and GPT-5.6 Sol make only 12, 7, 4, and 1 calls, respectively, across 36 tasks each (Table 3). MCR ranges from 0.91 to 1.00 across all systems, indicating that at least 91% of required clarification slots remain unresolved. Agents thus advance professional tasks while rarely seeking the information needed to meet user-specific requirements.

Preference-related delivery remains incomplete despite task progress. With DeepSeek-V4.1-Flash, DeepSeek Harness raises PTS from 0.107 to 0.218 and PCR from 0.032 to 0.063 (Table 2). Its PCR is the highest among evaluated systems but remains low, with all model baselines below 0.05. PCR counts full preference-related delivery checkpoints, including artifact and evidence conditions. Together with MCR, it reveals incomplete delivery and unresolved information needs beyond the task score. Section 4.2 examines how information access affects preference compliance.

Table 3: Execution statistics. Steps and kTok. report mean steps per task and thousand tokens per step, respectively. #Ask counts all `ask_user` calls across the 36 Profile tasks.

	Expert		Profile			Revision	
System	Steps	kTok.	Steps	kTok.	#Ask	Steps	kTok.
Agent Harness							
DeepSeek Harness							
+ DeepSeek-V4.1-Flash	110.6	134.2	117.3	135.3	4	182.2	205.4
Proprietary models							
GPT-5.4	107.7	65.5	101.0	38.6	12	146.0	80.1
GPT-5.6 Sol	110.1	35.9	111.6	34.8	1	217.9	45.5
GPT-5.6 Luna	138.5	55.7	145.7	56.0	0	245.6	68.7
Seed-2.1-Pro	133.3	63.0	158.6	86.9	0	219.4	94.0
Qwen3.7-Plus	81.1	59.6	59.5	66.1	0	103.7	70.1
Open-weight models							
DeepSeek-V4.1-Flash	104.4	62.5	108.0	78.2	7	191.6	110.7
GLM5.3-flash	67.6	44.4	50.7	50.6	0	95.9	67.5
Kimi K3	18.1	50.7	27.2	45.0	0	18.2	31.4
MiniMax M3	131.8	89.2	148.2	120.4	0	163.0	132.5

Table 4: Round-wise revision results. Ep.: episodes contributing revision units in rounds 1/2/3. FCR and CPR pool eligible units per round; full counts in Table 18.

	Ep.	Round 1		Round 2		Round 3	
System	(1/2/3)	FCR	CPR	FCR	CPR	FCR	CPR
Agent Harness							
DeepSeek Harness							
+ DeepSeek-V4.1-Flash	29/22/5	0.224	0.933	0.094	0.958	0.250	1.000
Proprietary models							
GPT-5.4	9/3/0	0.120	0.714	0.000	0.600	–	–
GPT-5.6 Sol	20/11/2	0.093	1.000	0.174	0.909	0.000	1.000
GPT-5.6 Luna	16/7/2	0.050	0.800	0.067	0.667	0.333	1.000
Seed-2.1-Pro	17/9/2	0.026	0.636	0.000	0.444	0.000	0.000
Qwen3.7-Plus	28/21/6	0.030	1.000	0.060	0.889	0.091	1.000
Open-weight models							
DeepSeek-V4.1-Flash	26/18/6	0.155	0.917	0.073	1.000	0.091	0.857
GLM5.3-flash	28/21/6	0.094	0.857	0.020	0.938	0.091	1.000
Kimi K3	5/2/0	0.000	–	0.000	–	–	–
MiniMax M3	21/16/4	0.022	0.333	0.000	–	0.000	–

3.4 Iterative Deliverable Revision

Agents can complete feedback rounds and preserve correct content while satisfying few feedback requirements. DeepSeek Harness completes all feedback rounds in 35/36 episodes but satisfies only 22/128 feedback requirements (FCR 0.172; Table 2). For DeepSeek-V4.1-Flash, the harness raises FRS from 0.068 to 0.116. GPT-5.6 Sol has limited evaluation coverage but a relatively high FCR (0.118) on evaluated units (Table 4). Qwen3.7-Plus preserves 22/23 previously correct units (CPR 0.957), but satisfies only 6/127 feedback requirements (FCR 0.047). Its PS of 0.215 shows that most protected requirements remain unmet. High CPR thus reflects retention of previously correct content, while low FCR and PS show that the revised deliverable still fails to meet most evaluated requirements.

LifelongCUAWorld shows that agents earn partial credit for professional work, but rarely seek missing user information and leave most evaluated feedback requirements unmet. Its collaboration metrics expose these gaps alongside overall task scores. Repeated evaluation with the same judge yields 95.6% verdict agreement and changes suite scores by less than 0.005, supporting repeatability under the fixed judge (Appendix I.5). The evaluated systems have yet to combine professional execution and sustained user collaboration into reliable lifelong assistance.

4 Analysis

We analyze how execution interfaces and CLI knowledge affect performance in Expert, and how user information affects preference-related delivery in Profile. For Revision, we vary feedback staging, workspace persistence, and draft source. All ablations use GLM5.3-flash on fixed subsets of 36 Expert tasks, 18 Profile tasks, and 18 Revision episodes, with tasks and budgets held constant within each comparison. All ablations retain the decoding parameters, reasoning effort, and evaluator configuration of the main experiments (Section 3.1; Appendix I.1).

4.1 Execution Interfaces and CLI Knowledge

Task-relevant CLI knowledge improves hybrid execution. CLI-only execution achieves higher scores in fewer steps than GUI-only execution (Table 5). Like function calls, CLI actions use explicit commands and arguments, reducing reliance on visual grounding. Hybrid execution also benefits from CLI knowledge: task-relevant registries (S2→S3) raise the score from 0.076 to 0.114 and CLI F1 to 0.377 with similar token use per step. All 17 registries (S5) yield similar scores but require 62% more tokens per step, supporting the task-relevant default. Task reference commands (S4) yield the highest score (0.142) and fewest steps among hybrid settings (58.6), showing that precise command guidance improves hybrid execution (Appendix I.2).

Table 5: Interface and CLI-knowledge ablation. S0: GUI only; S1: CLI only; S2: GUI and CLI without CLI knowledge; S3: S2 + task-relevant application registries (default); S4: S2 + task-specific reference commands; S5: S2 + all 17 application registries. kTok.: 10^3 tokens.

	Interface		CLI-KB					CLI diagnostics↑				
Setting	GUI	CLI	App	GT	All	**Score↑**	**Pass↑**	P	R	F1	**Steps**	**kTok./step**
S0	✓					0.000	0.0	0.000	0.000	0.000	123.6	**19.8**
S1		✓				0.120	0.0	0.000	0.000	0.000	**46.4**	39.2
S2	✓	✓				0.076	0.0	0.018	0.003	0.006	61.1	45.2
S3	✓	✓	✓			0.114	0.0	**0.457**	**0.373**	**0.377**	67.6	44.4
S4	✓	✓		✓		**0.142**	0.0	0.448	0.361	0.366	58.6	44.2
S5	✓	✓			✓	0.120	0.0	0.375	0.333	0.303	61.2	72.0

Table 6: User-information ablation. P_u: profile description; H_u: history; $Q_u^{<j}$ / $Q_{u,j}$: oracle Q/A of prior / current tasks.

P_u	H_u	$Q_u^{<j}$	$Q_{u,j}$	**PTS↑**	**PCR↑**
				0.104	0.056
✓				0.114	0.167
	✓			0.109	0.083
			✓	0.136	0.130
		✓	✓	**0.142**	**0.194**
	✓	✓	✓	0.126	0.139

Table 7: Revision protocol ablation. W0: full workspace carryover; W1: prior artifact only; W2: reset; $D^{(0)*}$: human-authored canonical draft.

Condition	**DFR↓**	**FRS↑**	**FCR↑**	**PS↑**	**CPR↑**
Staged, W0, agent draft *(default)*	1.000	0.052	0.074	0.309	0.917
Instruction					
Full instruction at once	–	0.023	–	–	–
Workspace					
W1: prior artifact file only	1.000	0.038	0.085	0.333	0.864
W2: reset (request + feedback)	1.000	0.043	0.071	0.165	0.421
Draft					
Oracle canonical draft $D^{(0)*}$	**0.000**	**0.183**	**0.175**	**0.635**	**0.956**

4.2 User Information and Preference Compliance

User information improves preference compliance. We disable online `ask_user` and vary the user information supplied before execution (Table 6). The five conditions use a task-conditioned description of P_u, user history H_u, or human-authored oracle Q/A. Oracle Q/A covers the current task alone or both current and preceding tasks; the combined Q/A is also evaluated with H_u. All five conditions improve both PTS and PCR over the instruction-only baseline (0.104 and 0.056, respectively). Disclosing P_u raises PCR to 0.167, while history alone raises it to 0.083, showing that both sources provide useful information for preference-compliant execution. With current-task Q/A fixed, adding prior oracle Q/A raises PCR from 0.130 to 0.194 and yields the highest PTS (0.142). Adding H_u to both Q/A sets lowers PCR to 0.139. These results show sensitivity to supplied user information and cross-task dependencies in LifelongCUAWorld-Profile (Appendix I.3).

4.3 Revision Protocol Ablations

Revision metrics provide complementary feedback and preservation diagnostics. Staged feedback yields higher FRS than full instruction (0.052 vs. 0.023) and provides snapshots for per-round evaluation (Table 7). Retaining only preceding artifacts (W1) yields CPR 0.864, versus 0.917 with full workspace carryover (W0). Resetting the workspace (W2) reduces CPR to 0.421, while FCR stays close to W0 (0.071 vs. 0.074). These conditional rates respond differently, supporting complementary feedback and preservation diagnostics. With feedback and evaluation unchanged, a human-authored canonical draft passing the initial-stage check improves FCR and FRS. FCR reaches only 0.175, leaving most evaluated feedback requirements unmet (Appendix I.4).

4.4 Case Study

Figure 5 examines consistency across a professional delivery package. GLM5.3-flash corrects the score's pagination and preserves approved notation, but leaves handout references and release records incomplete (FCR $1/4$; CPR $2/2$). Unit checks locate these unmet requirements alongside the preserved content, showing that professional delivery remains essential throughout user collaboration. Appendix J links requirements, artifact evidence, and evaluation decisions in all three suites.

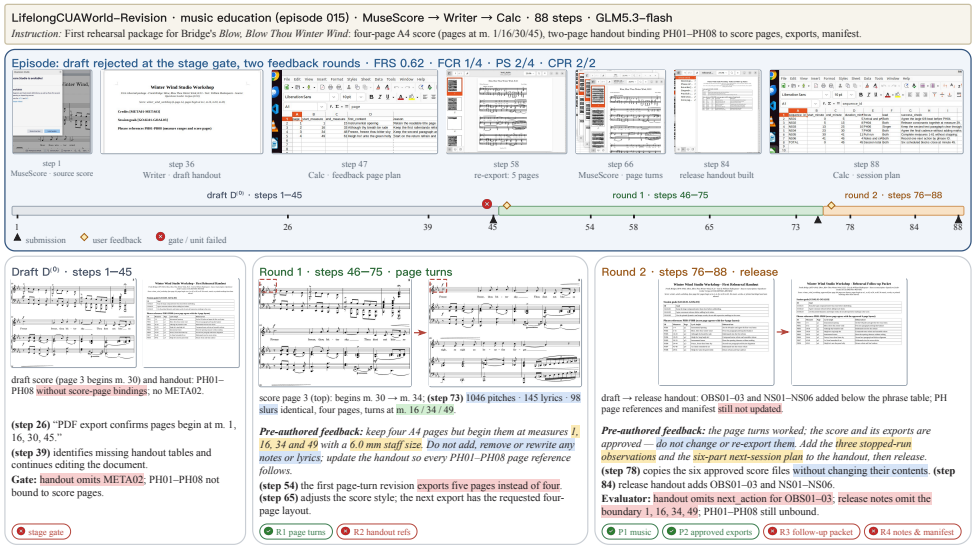


Figure 5: Revision case study. GLM5.3-flash preserves approved notation but leaves handout references and release records incomplete (yellow: feedback; blue: protected; green: achieved; red: failed).

5 Related Work

From task execution to professional workflows. Agent evaluation spans reasoning, web interaction, and software engineering (Liu et al., 2024; Deng et al., 2023; Zhou et al., 2024; Jimenez et al., 2024; Phan et al., 2025). Desktop and mobile benchmarks bring evaluation into real applications (Xie et al., 2024; Deng et al., 2024; Wang et al., 2024), supporting the development and assessment of computer-use models (Liu et al., 2025; Company, 2026). Recent benchmarks increasingly emphasize professional workflows, longer execution sequences, and dependencies across tools (Sun et al., 2025; Shi et al., 2026; CocoaBench Team et al., 2026; Yuan et al., 2026). Hybrid interfaces broaden how agents carry out these workflows (Zhao et al., 2025; Li et al., 2026), while verifiable professional deliverables provide a basis for assessing task outcomes (Sun et al., 2026).

From fixed instructions to user interaction. Alongside richer tasks, evaluation increasingly considers user context and interaction. Personalized environments and historical records provide information about individual users, supporting evaluation of preference inference and proactive assistance (Jang et al., 2026; Chai et al., 2026). Interactive tasks also require agents to acquire missing information and clarify user needs through dialogue (He et al., 2025; Kong et al., 2026; Chen et al., 2026). Feedback during execution or after an initial submission further brings user collaboration into professional desktop evaluation (Wang et al., 2026). These developments broaden the focus from executing a given instruction to using user information and responding to input as work proceeds. Table 8 in the appendix summarizes representative benchmarks and their evaluation protocols.

6 Conclusion

We introduce LifelongCUAWorld, a benchmark of 180 professional tasks for expert-level execution, cross-task preference learning, and feedback-driven revision. It combines artifact-based evaluation with dedicated collaboration metrics.

The evaluated systems show partial professional competence, and a suitable Agent Harness enables further progress, but complete delivery remains difficult. Agents rarely seek missing user information, and their revised deliverables satisfy few feedback requirements even when they complete feedback rounds and preserve correct content. Ablations support the CLI knowledge base and evaluation design: task-relevant CLI knowledge improves Expert scores, supplied prior user information improves Profile compliance, and Revision metrics respond differently to workspace conditions. Taken together, these findings highlight the remaining gap between current systems and reliable *lifelong computer-use agents*. LifelongCUAWorld provides a systematic evaluation platform to advance agents that understand users, collaborate on professional tasks, and learn across interactions.

AI USE STATEMENT

Generative AI tools assisted with research-code development, implementation-to-manuscript consistency checks, literature discovery, and manuscript drafting and revision. The evaluation framework of LifelongCUAWorld also uses LLM judges for rubric-based assessment of submitted artifacts, alongside deterministic rule checks (Section 2.5; Appendix G). Reported metrics are computed by evaluation code from execution records, rule-check results, and recorded judge assessments. The authors verified the implementation, claims, citations, figures, numerical results, and final text, and take full responsibility for them.

REFERENCES

Pranjal Aggarwal, Graham Neubig, and Sean Welleck. Gym-Anything: Turn any software into an agent environment. *arXiv preprint arXiv:2604.06126*, 2026.

Yuxiang Chai, Shunye Tang, Han Xiao, Rui Liu, and Hongsheng Li. PIRA-Bench: A transition from reactive GUI agents to GUI-based proactive intent recommendation agents. *arXiv preprint arXiv:2603.08013*, 2026.

Tongbo Chen, Zhengxi Lu, Zhan Xu, Guocheng Shao, Shaohan Zhao, Fei Tang, Yong Du, Kaitao Song, Yizhou Liu, Yuchen Yan, et al. KnowU-Bench: Towards interactive, proactive, and personalized mobile agent evaluation. *arXiv preprint arXiv:2604.08455*, 2026.

CocoaBench Team, Shibo Hao, Zhining Zhang, Zhiqi Liang, Tianyang Liu, Yuheng Zha, Qiyue Gao, Jixuan Chen, Zilong Wang, Zhoujun Cheng, et al. CocoaBench: Evaluating unified digital agents in the wild. *arXiv preprint arXiv:2604.11201*, 2026.

H Company. Holo3 - open foundation models for navigation and computer use agents, 2026. URL https://huggingface.co/Hcompany/Holo3-35B-A3B.

DeepSeek-AI. Deepseek harness: Everything is a plugin. https://github.com/deepseek-ai/deepseek-harness, 2026a.

DeepSeek-AI. Deepseek-v4.1-flash: Pushing the limits of kv cache compression, 2026b. URL https://arxiv.org/abs/2609.19969.

Shihan Deng, Weikai Xu, Hongda Sun, Wei Liu, Tao Tan, Jianfeng Liu, Ang Li, Jian Luan, Bin Wang, Rui Yan, and Shuo Shang. Mobile-Bench: An evaluation benchmark for LLM-based mobile agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8813–8831, 2024. doi: 10.18653/v1/2024.acl-long.478. URL https://aclanthology.org/2024.acl-long.478/.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

GLM-5-Team. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Wei He, Yueqing Sun, Hongyan Hao, Xueyuan Hao, Zhikang Xia, Qi Gu, Chengcheng Han, Dengchang Zhao, Hui Su, Kefeng Zhang, et al. VitaBench: Benchmarking LLM agents with versatile interactive tasks in real-world applications. *arXiv preprint arXiv:2509.26490*, 2025.

Lawrence Keunho Jang, Andrew Keunwoo Jang, Jing Yu Koh, and Ruslan Salakhutdinov. MyPCBench: A benchmark for personally intelligent computer-use agents. *arXiv preprint arXiv:2606.16748*, 2026.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Kimi Team. Kimi k3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

Quyu Kong, Xu Zhang, Zhenyu Yang, Nolan Gao, Chen Liu, Panrong Tong, Chenglin Cai, Hanzhang Zhou, Jianan Zhang, Liangyu Chen, Zhidan Liu, Steven Hoi, and Yue Wang. MobileWorld: Benchmarking autonomous mobile agents in agent-user interactive and MCP-augmented environments. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6142–6167, 2026. doi: 10.18653/v1/2026.acl-long.278. URL https://aclanthology.org/2026.acl-long.278/.

Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, Yang Xu, Lunbin Zeng, Xiaolong Li, Haohai Sun, Haichao Zhu, Vito Zhang, et al. Minimax sparse attention. *arXiv preprint arXiv:2606.13392*, 2026.

Wanli Li, Bowen Zhou, Yunyao Yu, Zhou Xu, Yifan Yang, Dongsheng Li, and Caihua Shan. WeaveBench: A long-horizon, real-world benchmark for computer-use agents with hybrid interfaces. *arXiv preprint arXiv:2606.09426*, 2026.

Guangyi Liu, Pengxiang Zhao, Yaozhen Liang, Liang Liu, Yaxuan Guo, Han Xiao, Weifeng Lin, Yuxiang Chai, Yue Han, Shuai Ren, et al. LLM-powered GUI agents in phone automation: Surveying progress and prospects. *arXiv preprint arXiv:2504.19838*, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zAdUB0aCTQ.

OpenAI. GPT-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Qwen Team. Qwen3.7-Plus: Multimodal agent intelligence, May 2026. URL https://qwen.ai/blog?id=qwen3.7-plus.

Chris Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, volume 2025, pp. 406–441, 2025.

Kean Shi, Zihang Li, Tianyi Ma, Zengji Tu, Jialong Wu, Xinbo Xu, Qingyao Yang, Ruoyu Wu, Weichu Xie, Ming Wu, et al. SaaS-Bench: Can computer-use agents leverage real-world SaaS to solve professional workflows? *arXiv preprint arXiv:2605.15777*, 2026.

Qiushi Sun, Zhoumianze Liu, Chang Ma, Zichen Ding, Fangzhi Xu, Zhangyue Yin, Haiteng Zhao, Zhenyu Wu, Kanzhi Cheng, Zhaoyang Liu, et al. ScienceBoard: Evaluating multimodal autonomous agents in realistic scientific workflows. *arXiv preprint arXiv:2505.19897*, 2025.

Yiyou Sun, Xinyang Han, Weichen Zhang, Yuanbo Pang, Tianyu Wang, Yuhan Cao, Yixiao Huang, Chris Duroiu, Haoyun Zhang, Jeffrey Lin, et al. Agents' last exam. *arXiv preprint arXiv:2606.05405*, 2026.

Fei Tang, Haolei Xu, Hang Zhang, Siqi Chen, Xingyu Wu, Yongliang Shen, Wenqi Zhang, Guiyang Hou, Zeqi Tan, Yuchen Yan, et al. A survey on (M)LLM-based GUI agents. *arXiv preprint arXiv:2504.13865*, 2025.

Luyuan Wang, Yongyu Deng, Yiwei Zha, Guodong Mao, Qinmin Wang, Tianchen Min, Wei Chen, and Shoufa Chen. MobileAgentBench: An efficient and user-friendly benchmark for mobile LLM agents. *arXiv preprint arXiv:2406.08184*, 2024.

Wenkai Wang, Tao Xiong, Jingchen Ni, Yunpeng Bao, Xiyun Li, Tianqi Liu, Hongcan Guo, Zilong Huang, and Shengyu Zhang. DeskCraft: Benchmarking desktop agents on professional workflows and human-in-the-loop collaboration. *arXiv preprint arXiv:2606.03103*, 2026.

Jinbiao Wei, Qianran Ma, Yilun Zhao, Xiao Zhou, Kangqi Ni, Guo Gan, and Arman Cohan. OpenComputer: Verifiable software worlds for computer-use agents. *arXiv preprint arXiv:2605.19769*, 2026.

Jingxu Xie, Dylan Xu, Xuandong Zhao, and Dawn Song. AgentSynth: Scalable task generation for generalist computer-use agents. *arXiv preprint arXiv:2506.14205*, 2025.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing*

Systems, 37:52040–52094, 2024.

Yuhao Yang, Tianyu Fan, and Chao Huang. CLI-Anything: Towards agent-native computer use, 2026. URL https://arxiv.org/abs/2606.03854.

Mengqi Yuan, Zilong Zhou, Xinzhuang Xiong, Weiming Wu, Jiayang Sun, Jiamin Song, Kaiqian Cui, Bowen Wang, Haoyuan Wu, Yitong Li, et al. OSWorld 2.0: Benchmarking computer use agents on long-horizon real-world tasks. *arXiv preprint arXiv:2606.29537*, 2026.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. AppAgent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–20, 2025.

Pengxiang Zhao, Guangyi Liu, Yaozhen Liang, Weiqing He, Zhengxi Lu, WenHao Wang, Yuehao Huang, Yuxiang Chai, Zhaolu Kang, Yaxuan Guo, et al. MAS-Bench: A unified benchmark for shortcut-augmented hybrid mobile GUI agents. *arXiv preprint arXiv:2509.06477*, 2025.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. GPT-4V(ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rmiwIL98uQ.

Contents of the appendix

A Limitations and Future Work 12
B Benchmark Comparison 12
C Statistics and Evaluation Subsets 13
C.1 Benchmark Composition 13
C.2 Evaluation Metrics at a Glance 13
C.3 Expert Subset and Step Budgets 14
C.4 Token and Cost Accounting 14
C.5 Profile Subset and Detailed Results 14
C.6 Revision Subset and Detailed Results . . . 17
D Professional Scenario Construction 18
D.1 Materials and Software Resources 18
D.2 Expert Personas and Task Derivation . . . 18
E Profile Construction and Interaction 18
E.1 Reference Corpus 18
E.2 Professional Role Tracks 19
E.3 User Information and Interaction 19
E.4 Profile Metrics 19
F Revision Protocol Details 20
F.1 Episode Structure and Feedback Annotations 20
F.2 Stage-Wise Evaluation and Coverage . . . 20
G Evaluator Details 20
G.1 Two-Layer Configuration 20
G.2 Artifact Evidence Extraction 21
G.3 Workflow Diagnostics 21
H Task and Evaluator Validation 21
H.1 Task Verification 21
H.2 Evaluator Validation 21
H.3 Fail-Condition Audit and Veto Policy . . 21
H.4 Profile and Revision Audits 22
I Supplementary Experiments 22
I.1 Model Settings 22
I.2 Interfaces and CLI Knowledge 23
I.3 User Information and Preferences 23
I.4 Revision Protocol and Persistence 23
I.5 Content-Judge Stability 23
J Case Studies 25
J.1 Expert Case Study 25
J.2 Profile Case Study 26
J.3 Revision Case Study 27

A Limitations and Future Work

Main results cover representative subsets and bounded sequences. The tested content checkpoints show repeatability under a fixed judge (Appendix I.5); independent expert annotations could further assess judgment validity. We hope LifelongCUAWorld will provide a platform for developing lifelong CUAs that learn users' needs and preferences over time, collaborate with users on professional tasks, and provide reliable, sustained assistance.

B Benchmark Comparison

Table 8 compares representative benchmarks on professional execution, user-profile modeling, and deliverable revision.

Table 8: Comparison of representative computer-use benchmarks on professional execution, user-profile modeling, and deliverable revision.

Benchmark	Scale	Professional Execution			User Profile Modeling			Deliverable Revision		
		Professional Tasks	Hybrid Operations	Deliverable Evaluation	Preference Learning	Proactive AskUser	Profile Accumulation	User Feedback	Revision Scoring	Content Preservation
Human-designed professional CUA benchmarks										
OSWorld 2.0 (Yuan et al., 2026)	108	✓	✓	✓	✗	✓	✗	✗	✗	✗
Agents' Last Exam (Sun et al., 2026)	1K+	✓	✓	✓	✗	✗	✗	✗	✗	✗
WeaveBench (Li et al., 2026)	114	✓	✓	✓	✗	✗	✗	✗	✗	✗
Large-scale synthetic CUA benchmarks										
AgentSynth (Xie et al., 2025)	6K+	✗	✗	✗	✗	✗	✗	✗	✗	✗
Gym-Anything (Aggarwal et al., 2026)	10K+	✓	✗	✗	✗	✗	✗	✗	✗	✗
OpenComputer (Wei et al., 2026)	1K	✗	✗	✓	✗	✗	✗	✗	✗	✗
User-aware and interactive GUI benchmarks										
MobileWorld (Kong et al., 2026)	201	✗	✓	✗	✗	✓	✗	✗	✗	✗
MyPCBench (Jang et al., 2026)	184	✗	✓	✗	✓	✗	✗	✗	✗	✗
KnowU-Bench (Chen et al., 2026)	192	✗	✗	✗	✓	✓	✗	✗	✗	✗
DeskCraft (Wang et al., 2026)	538	✓	✗	✓	✗	✓	✗	✓	✗	✗
Lifelong CUA benchmark										
LifelongCUAWorld (Ours)	**180**	✓	✓	✓	✓	✓	✓	✓	✓	✓

Checks indicate dedicated protocol or evaluation features. Profile Accumulation concerns retained interactions across tasks for the same user; Revision Scoring concerns feedback evaluation against successive artifact snapshots; Content Preservation concerns explicit scoring of protected content. A cross denotes absence of that dedicated feature, not absence of related behavior during execution.

C Benchmark Statistics and Evaluation Subsets

C.1 Benchmark Composition

LifelongCUAWorld contains 180 tasks: 72 Expert, 54 Profile, and 54 Revision. Every task starts from a standardized workspace with input assets, project files, and reference materials. The Expert suite spans 13 professional domains and 19 desktop applications, from CAD, GIS, and 3D design to audio and video production, diagramming, game prototyping, and knowledge management; 66 of its 72 tasks (92%) involve two or more applications. Table 1 in the main text contrasts the evaluation unit, the visible user context, and the state persistence of the three suites.

C.2 Evaluation Metrics at a Glance

Table 9 summarizes the metrics used across the three suites, including their interpretation and aggregation. It distinguishes overall deliverable quality from preference compliance, clarification behavior, feedback compliance, and content preservation.

Table 9: Evaluation metrics by suite. All scores and rates lie in $[0, 1]$; arrows indicate the preferred direction. Task scores credit partial progress, whereas Pass records strict completion.

Metric	Meaning	Definition and aggregation
Expert		
Score↑	Deliverable quality	Mean task score $S(\tau)$ from the rule and content layers, including the task's completion and validity factor (Eq. 1).
Pass↑	Strict task completion	Fraction of tasks meeting the score threshold and passing every critical check.
CLI P/R/F1↑	Command-use diagnostics	Precision, recall, and F1 of issued commands against task reference commands; reported separately from the task score.
Profile		
PTS↑	Profile Task Score	Mean $S(\tau)$ across all evaluated Profile tasks.
PCR↑	Preference Compliance Rate	Weighted satisfaction of preference-related delivery checkpoints, including specified artifact and evidence conditions; weights are normalized within each task before pooling.
RAR↓	Redundant Ask Rate	Redundant `ask_user` calls divided by all calls; undefined when the agent makes no query.

Continued on next page.

Table 9: Evaluation metrics by suite (continued).

Metric	Meaning	Definition and aggregation
MCR↓	Missed Clarification Rate	Unanswered task-critical slots divided by the slots requiring clarification, pooled across tasks. The denominator excludes inapplicable or already-resolved slots.
Revision		
DFR↓	Draft Failure Rate	Drafts failing the initial-stage acceptance check divided by observable drafts.
FRS↑	Final Revision Score	Mean final deliverable score $S(D^{(K)})$ over all evaluated episodes.
FCR↑	Feedback compliance	$\lvert\mathcal{R}^{+}\rvert/\lvert\mathcal{R}\rvert$: feedback requirements satisfied after revision divided by eligible feedback requirements, including explicit preservation constraints.
PS↑	Preservation requirements	$\lvert\mathcal{P}^{+}\rvert/\lvert\mathcal{P}\rvert$: preservation units valid after revision divided by all eligible preservation units, whether or not they were valid before revision.
CPR↑	Retention of previously correct content	$\lvert\mathcal{P}^{\circ+}\rvert/\lvert\mathcal{P}^{\circ}\rvert$: previously valid preservation units that remain valid after revision; undefined if none were valid beforehand.

FCR, PS, and CPR pool eligible units over reached feedback rounds and require coverage-aware interpretation. Feedback and preservation units may concern overlapping properties; FCR scores post-revision satisfaction rather than before-to-after transitions. Detailed definitions appear in Appendices E.4, F.2, and G.3.

C.3 Expert Subset and Step Budgets

The Expert experiments and ablations use a fixed subset of 36 representative tasks covering all 13 domains and 19 applications (Table 10). All systems use a per-task step limit of twice the estimated step count (88–250 steps; median 148).
Performance varies widely across domains (Table 11). DeepSeek Harness achieves the highest overall score (0.239), with its strongest result in score engraving (0.56). All systems receive partial credit in visual identity design, but none receives credit in knowledge management. No evaluated system strictly completes an Expert task (Table 2).

Table 10: Domain coverage of the 36-task Expert subset (tasks selected / tasks in the 72-task suite) and its step budgets (min–max).

Professional domain	Tasks	Step budget
Academic literature and research	4 / 7	96–156
Architecture and 3D visualization	4 / 7	128–156
Software and systems architecture	4 / 7	88–152
Data analysis and reporting	2 / 5	144–156
Music education	2 / 5	144–148
Game prototyping	2 / 5	176–250
Geospatial analysis	3 / 6	152–172
Knowledge management	2 / 4	148–164
Live production and streaming	2 / 3	148–156
Mechanical design	3 / 7	104–164
Audio and video production	4 / 8	124–148
Music notation and rehearsal	2 / 4	144–164
Visual identity and design	2 / 4	116–136
Total	**36 / 72**	**88–250**

C.4 Token and Cost Accounting

Tokens per step are calculated by dividing total input and output tokens by total execution steps. We estimate mean API cost per task from the recorded token usage and the list prices in Table 12. For DeepSeek Harness, we count native SDK steps across sessions and include cache reads and writes in input-token usage.

C.5 Profile Subset and Detailed Results

The Profile subset contains six tasks from each of the six role tracks, covering 16 applications and the discovery, reuse, and adaptation stages (Table 14). Tasks retain their original order. The workspace resets between tasks, while H_u and the agent's prior exchanges $A_{a,u}^{<j}$ persist. Selection preserves cross-task dependencies: every reused preference can be elicited in an earlier selected task for the same user. The subset includes the 18 tasks used in the information ablation of Section 4.2. All systems use a per-task step limit of twice the estimated step count (112–312 steps; median 166).

Table 11: Expert task score by domain (36-task subset; n tasks per domain in the second header row). Darker green indicates higher scores; the overall **best** and second-best scores are highlighted.

System / n	Lit. review 4	Arch. viz. 4	SW arch. 4	Sales dash. 2	Music lesson 2	Game proto. 2	GIS analysis 3	Knowl. mgmt 2	Live stream 2	Mech. design 3	Podcast video 4	Score engr. 2	Brand ident. 2	All domains 36
Agent Harness														
DeepSeek Harness + DeepSeek-V4.1-Flash	0.23	0.22	0.11	0.17	0.39	0.21	0.32	0	0.29	0.27	0.22	0.56	0.25	**0.239**
Proprietary models														
GPT-5.4	0	0.10	0.14	0.08	0.46	0	0.14	0	0	0.10	0.08	0.22	0.33	0.116
GPT-5.6 Sol	0	0.11	0.12	0.17	0.21	0.16	0.11	0	0	0.06	0.08	0	0.35	0.097
GPT-5.6 Luna	0	0.06	0	0	0.33	0	0.22	0	0	0	0	0	0.31	0.061
Seed-2.1-Pro	0	0.03	0	0.08	0.21	0	0.17	0	0.09	0	0	0.26	0.25	0.067
Qwen3.7-Plus	0.08	0.07	0	0.13	0.21	0.10	0	0	0	0.04	0.04	0.09	0.08	0.058
Open-weight models														
DeepSeek-V4.1-Flash	0.07	0.04	0.14	0.08	0.38	0.16	0.15	0	0	0.04	0	0.26	0.09	0.098
GLM5.3-flash	0.07	0.12	0.06	0.13	0.44	0.10	0.03	0	0.09	0.16	0.06	0.22	0.17	0.114
Kimi K3	0	0.04	0	0.08	0	0	0	0	0	0	0	0.26	0.09	0.028
MiniMax M3	0	0.06	0.02	0.09	0	0	0	0	0	0	0	0.22	0.09	0.030

Table 12: List prices (USD per million tokens) for standard uncached requests in the base context tier, September 2026. GPT-5.6 Sol is listed at its standard price; DeepSeek uses peak rates.

System	Provider	Input	Output	Base tier
Agent Harness				
DeepSeek Harness + DeepSeek-V4.1-Flash	DeepSeek	0.30	1.20	1M
Proprietary models				
GPT-5.4	OpenAI	2.50	15.00	$\leq$272K
GPT-5.6 Sol	OpenAI	5.00	30.00	$\leq$272K
GPT-5.6 Luna	OpenAI	0.20	1.20	$\leq$272K
Seed-2.1-Pro	ByteDance Volcano Engine	0.89	4.44	$\leq$256K
Qwen3.7-Plus	Alibaba Cloud Model Studio	0.40	1.60	$\leq$256K
Open-weight models				
DeepSeek-V4.1-Flash	DeepSeek	0.30	1.20	1M
GLM5.3-flash	Z.ai	0.15	0.50	–
Kimi K3	Moonshot AI	3.00	15.00	–
MiniMax M3	MiniMax	0.30	1.20	$\leq$512K

The 36 tasks define 95 task-critical clarification slots. Slots already resolved by answers retained from earlier tasks do not require repeated clarification. MCR measures the fraction of remaining required slots unresolved at task completion (Appendix E.4).

Task competence and preference compliance. The systems make progress on professional tasks but satisfy few complete preference-related delivery units. With DeepSeek-V4.1-Flash, DeepSeek Harness achieves PTS 0.218 and PCR 0.063, compared with 0.107 and 0.032 for the model baseline. In a broadcast-delivery task, GPT-5.4 obtains the required answers and applies the requested media behaviors, but misses associated project requirements (Appendix J.2). No evaluated system strictly completes a Profile task.

Clarification behavior. Only four systems invoke `ask_user`: GPT-5.4, DeepSeek-V4.1-Flash, DeepSeek Harness, and GPT-5.6 Sol make 12, seven, four, and one calls, respectively. Half of GPT-5.4's calls meet the call-level redundancy criterion, which also includes calls mixing redundant topics with useful questions. The other three systems make no redundant calls. These four systems still leave 91–98% of required clarification slots unresolved and satisfy few complete preference-related delivery units (PCR 0.028–0.063). The remaining six systems never ask, so RAR is undefined and MCR is 1.000.

Table 13: Mean input (M) and output (k) tokens per task and the resulting Cost/task (USD) of Table 2.

	Expert			Profile			Revision		
System	In (M)	Out (k)	Cost	In (M)	Out (k)	Cost	In (M)	Out (k)	Cost
Agent Harness									
DeepSeek Harness + DeepSeek-V4.1-Flash	14.734	109.6	4.55	15.731	140.9	4.89	37.209	215.8	11.42
Proprietary models									
GPT-5.4	6.835	218.5	20.36	3.876	23.0	10.04	11.409	276.8	32.67
GPT-5.6 Sol	3.927	29.4	20.52	3.852	28.7	20.12	9.864	55.4	50.98
GPT-5.6 Luna	7.607	107.9	1.65	8.068	98.8	1.73	16.719	150.3	3.52
Seed-2.1-Pro	8.342	55.3	7.66	13.706	78.8	12.53	20.541	87.7	18.65
Qwen3.7-Plus	4.787	49.0	1.99	3.881	49.6	1.63	7.180	89.6	3.02
Open-weight models									
DeepSeek-V4.1-Flash	6.181	344.9	2.27	7.972	471.8	2.96	20.637	572.4	6.88
GLM5.3-flash	2.913	85.9	0.48	2.460	101.3	0.42	6.324	152.5	1.02
Kimi K3	0.889	28.3	3.09	1.188	34.3	4.08	0.550	21.4	1.97
MiniMax M3	11.715	43.5	3.57	17.777	74.6	5.42	21.514	85.0	6.56

Table 14: Composition and step budgets of the 36-task Profile subset. Stages: discovery (D), reuse (R), and adaptation (A) tasks; budget range is the min–max within a track.

Role track	**Tasks**	**D/R/A**	**Step budget**
Research and Knowledge Synthesis	6	2/2/2	112–240
Data and Systems Decision-Making	6	2/2/2	116–240
Spatial Planning and the Built Environment	6	3/1/2	160–312
Mechanical Product Design	6	2/2/2	144–264
Brand and Interaction Experience	6	3/1/2	148–216
Music Education and Media Production	6	2/2/2	128–224
Total	**36**	**14/10/12**	**112–312**

Table 15: Role-level Profile results (PTS / PCR); six tasks per track. PTS measures overall task quality; PCR measures satisfaction of complete preference-related delivery units. Within each track, the **best** and second-best nonzero values are highlighted separately for PTS and PCR, including ties.

System	**Research**	**Data/Systems**	**Spatial**	**Mechanical**	**Brand**	**Music/Media**
Agent Harness						
DeepSeek Harness + DeepSeek-V4.1-Flash	**0.2833** / **0.2917**	**0.2413** / **0.0833**	**0.1802** / 0	**0.3454** / 0	**0.1918** / 0	0.0672 / 0
Proprietary models						
GPT-5.4	0.0613 / 0.1944	0.0126 / **0.0833**	0.0318 / 0	0.0705 / 0	0.0200 / 0	0.0698 / 0
GPT-5.6 Sol	0.1099 / 0.1667	0.0835 / 0	0.0109 / 0	0.0802 / 0	0.0167 / 0	**0.0706** / 0
GPT-5.6 Luna	0.0721 / 0.1667	0.1258 / 0	0.0255 / 0	0.1157 / 0	0.0467 / 0	0.0000 / 0
Seed-2.1-Pro	0.1515 / **0.2917**	0.1535 / 0	0.1600 / 0	0.0809 / 0	0.0739 / 0	0.0000 / 0
Qwen3.7-Plus	0.0467 / 0.0556	0.0743 / 0	0.0454 / 0	0.0818 / 0	0 / 0	0.0367 / 0
Open-weight models						
DeepSeek-V4.1-Flash	0.1404 / 0.1944	0.0433 / 0	0.0953 / 0	0.2051 / 0	0.1355 / 0	0.0200 / 0
GLM5.3-flash	0.1713 / **0.2917**	0.1342 / 0	0.1101 / 0	0.1379 / 0	0.0367 / 0	0.0367 / 0
Kimi K3	0.1217 / 0.2500	0.0543 / 0	0.0574 / 0	0.0159 / 0	0 / 0	0.0567 / 0
MiniMax M3	0.1153 / 0.2083	0.0659 / 0	0.0529 / 0	0.0870 / 0	0 / 0	0 / 0

Table 3 reports execution steps and token use; Table 13 gives input/output tokens and estimated costs.

C.6 Revision Subset and Detailed Results

The main Revision experiments use 36 episodes, four from each of nine scenarios (Table 16). Episodes contain one to three feedback rounds, with 68 rounds in total. Each of six feedback types appears in six episodes: quality refinement, scope extension, parameter correction, delivery update, preference override, and constraint reconciliation. All systems use a per-episode step limit of twice the estimated step count (176–472 steps).

Table 16: Composition of the 36-episode Revision subset. Rounds: episodes with one, two, and three feedback rounds.

Scenario	Episodes	Rounds (1/2/3)
Research and Knowledge Management	4	1/2/1
Data and Systems Decision-Making	4	1/2/1
GIS	4	1/2/1
Mechanical Design	4	1/2/1
Architecture Visualization	4	1/2/1
Brand and Visual Identity	4	1/2/1
Game Prototyping	4	1/2/1
Music Education	4	3/1/0
Media and Live Streaming	4	2/1/1
Total	**36**	**12/16/8**

Protocol completion and metric coverage. Completing the feedback sequence and satisfying its requirements are distinct outcomes. DeepSeek Harness submits every feedback round in 35 of 36 episodes, yet satisfies only 22 of 128 evaluated feedback requirements (Table 17). All 36 episodes contribute to FRS. FCR and PS aggregate units with valid evaluation results from reached rounds; CPR further restricts its denominator to preservation units satisfied before revision. These conditional metrics should be read with their unit counts and stage coverage, rather than as standalone measures of end-to-end revision success. Table 18 reports the corresponding counts per round.

Table 17: Coverage on the 36 Revision episodes. All rounds: episodes with a submission after every feedback round, not strict task completions. DFR: failed / observable drafts. FCR and PS: satisfied / evaluated units; CPR: retained / previously satisfied preservation units.

System	All rounds	DFR	FCR	PS	CPR
Agent Harness					
DeepSeek Harness + DeepSeek-V4.1-Flash	35/36	33/35	22/128	58/134	43/45
Proprietary models					
GPT-5.4	9/36	23/24	3/31	12/30	8/12
GPT-5.6 Sol	17/36	35/36	8/68	25/57	21/22
GPT-5.6 Luna	14/36	35/35	4/58	19/61	11/14
Seed-2.1-Pro	19/36	34/35	1/63	16/65	11/22
Qwen3.7-Plus	34/36	36/36	6/127	29/135	22/23
Open-weight models					
DeepSeek-V4.1-Flash	28/36	35/36	13/110	39/115	32/34
GLM5.3-flash	34/36	34/35	8/124	37/131	32/35
Kimi K3	3/36	36/36	0/15	1/9	–
MiniMax M3	23/36	31/31	1/88	2/93	1/3

Compliance and preservation. PS measures whether designated protected requirements hold after revision; CPR restricts this assessment to those already satisfied beforehand. These denominators answer different questions: GPT-5.6 Luna has higher PS than GLM5.3-flash (0.311 vs. 0.282), but lower CPR (0.786 vs. 0.914). High CPR also need not imply high feedback compliance: Qwen3.7-Plus retains 22 of 23 previously valid preservation units but satisfies only six of 127 evaluated feedback requirements.

Table 18: Round-wise metric coverage for Table 4. Episodes: episodes contributing evaluated feedback units in that round. FCR and PS report satisfied / evaluated units; CPR reports retained / previously satisfied preservation units. A dash indicates no eligible units.

	Round 1				Round 2				Round 3			
System	Episodes	FCR	PS	CPR	Episodes	FCR	PS	CPR	Episodes	FCR	PS	CPR
Agent Harness												
DeepSeek Harness + DeepSeek-V4.1-Flash	29	15/67	21/42	14/15	22	5/53	28/71	23/24	5	2/8	9/21	6/6
Proprietary models												
GPT-5.4	9	3/25	9/19	5/7	3	0/6	3/11	3/5	0	–	–	–
GPT-5.6 Sol	20	4/43	11/24	9/9	11	4/23	11/25	10/11	2	0/2	3/8	2/2
GPT-5.6 Luna	16	2/40	14/34	8/10	7	1/15	4/19	2/3	2	1/3	1/8	1/1
Seed-2.1-Pro	17	1/39	10/28	7/11	9	0/22	4/29	4/9	2	0/2	2/8	0/2
Qwen3.7-Plus	28	2/66	13/43	9/9	21	3/50	10/67	8/9	6	1/11	6/25	5/5
Open-weight models												
DeepSeek-V4.1-Flash	26	9/58	13/37	11/12	18	3/41	19/53	15/15	6	1/11	7/25	6/7
GLM5.3-flash	28	6/64	14/39	12/14	21	1/49	16/67	15/16	6	1/11	7/25	5/5
Kimi K3	5	0/10	–	–	2	0/5	1/9	–	0	–	–	–
MiniMax M3	21	1/45	2/25	1/3	16	0/35	0/52	–	4	0/8	0/16	–

D Professional Scenario Construction

D.1 Professional Materials and Software Resources

Tasks draw on research papers, public datasets, technical guidance, media assets, and open-source projects collected online. We prioritize materials published or substantially updated within the two years before benchmark construction. Expert and Profile also use older standards and historical inputs where the task requires them. Each task records its sources, versions, licenses, and any transformations used to prepare the input assets.

For each application, we compile a capability handbook from official documentation, public tutorials, and open-source projects. The handbook describes professional uses, supported operations, inputs and outputs, and ways to verify task completion. CLI-Anything (Yang et al., 2026) extracts command-line capabilities into application registries with command templates, arguments, preconditions, and verification guidance. We use the handbooks to design tasks and provide task-relevant registries as the agent's CLI knowledge under the default configuration. Section 4.1 compares the CLI-knowledge conditions.

D.2 Expert Personas and Task Derivation

We use the professional materials to define expert roles and recurring workflows, including work objectives, deliverables, quality requirements, and review responsibilities. We then combine these workflows with application capabilities to specify task goals, initial workspaces, input assets, intermediate and final deliverables, and evaluation checks. Profile adds requirements that depend on user information; Revision adds staged feedback and requirements to preserve previously correct content.

E Profile Construction and Interaction Protocol

E.1 Reference Tasks for User Profile Construction

We use reference tasks from existing computer-use benchmarks to construct six user archetypes for LifelongCUAWorld-Profile (Table 19). Their professional contexts, software use, workflows, and preference requirements help identify recurring work habits and user attributes. From these archetypes, we author the profiles, histories, task sequences, interaction slots, and preference checkpoints for LifelongCUAWorld.

Table 19: Reference task sources for user profile construction.

Source	Evidence for User Profiles
MyPCBench (Jang et al., 2026)	Personal context, work habits, and user-information dependencies
OSWorld 2.0 (Yuan et al., 2026)	Software use, recurring work practices, and cross-application context
Agents' Last Exam (Sun et al., 2026)	Professional responsibilities, work products, and quality expectations

E.2 Professional Role Tracks

Each user archetype has a nine-task sequence designed from the collected professional materials (Table 20). Tasks within a track share a professional context and recurring preferences, allowing the agent to apply user information across different deliverables and work stages.

Table 20: Profile role tracks with nine tasks each; example workflows and preference categories.

Professional Role Track	Workflows and Deliverables	Recurring Preferences
Research and Knowledge Synthesis	Literature synthesis, reports, bibliographies, and knowledge bases	Citations, traceability, structure, and naming
Data and Systems Decision-Making	Data analysis, spreadsheets, reports, and dashboards	Validation, charts, thresholds, and export formats
Spatial Planning and the Built Environment	Spatial analysis, maps, GIS projects, and planning documents	CRS, layers, symbology, and annotation density
Mechanical Product Design	CAD models, assemblies, drawings, and bills of materials	Units, tolerances, naming, and release conventions
Brand and Interaction Experience	Visual identities, design assets, and interface prototypes	Colors, typography, layout, and delivery specifications
Music Education and Media Production	Scores, audio projects, video timelines, and instructional media	Notation, audio settings, timing, and publishing formats

E.3 User Information and Controlled Interaction

Each user's hidden profile P_u specifies stable facts, work habits, deliverable preferences, quality requirements, and contextual exceptions. The agent receives a natural-language history H_u and its own prior `ask_user` exchanges $A_{a,u}^{<j}$. For each task, annotations distinguish information available from history or earlier answers, information that requires clarification, and preferences superseded by the current request. Tasks run in a fixed order. The workspace resets between tasks, while H_u and prior exchanges carry forward.

The `ask_user` responder uses predefined topics, canonical answers, availability conditions, and refusal rules. It maps free-form questions to topics and returns the corresponding answer when the topic is recognized and currently answerable; otherwise, it returns a predefined refusal. Predefined answers and refusals reduce variability in user responses, supporting stable and reproducible evaluation. Some answerable topics are irrelevant to the current task, allowing the evaluator to distinguish necessary clarification from redundant questions.

An answer resolves later clarification requirements only if it applies to the current task and is marked for cross-task reuse. Refusals and task-local answers remain visible in the transcript but do not resolve later requirements. Each preference checkpoint records whether its required information comes from H_u, prior exchanges, or current-task answers.

E.4 Profile Evaluation Metrics

Let $\mathcal{T}_P$ denote the evaluated Profile tasks and $\mathcal{K}_\tau^{\text{pref}}$ the preference checkpoints of task τ. These checkpoints use deterministic rules or evidence-based content judgments. The **Profile Task Score**,

$\mathrm{PTS} = \frac{1}{|\mathcal{T}_P|} \sum_{\tau \in \mathcal{T}_P} S(\tau)$, averages the task scores defined in Eq. 1. The **Preference Compliance Rate**, $\mathrm{PCR} = \sum_{\tau,k} \omega_k q_k \,/\, \sum_{\tau,k} \omega_k$, aggregates preference-checkpoint scores $q_k \in [0,1]$, with weights ω_k normalized within each task. A checkpoint assesses the complete preference-related delivery unit, including its specified artifact and evidence requirements. PCR therefore measures satisfaction of these full units; it does not isolate preference understanding from professional delivery.

The **Redundant Ask Rate**, $\mathrm{RAR}_a = \sum_{q \in \mathcal{Q}_a} \rho(q)/|\mathcal{Q}_a|$, is the fraction of agent a's `ask_user` calls marked redundant ($\rho(q) = 1$). A call is redundant if any matched topic is already answered by the context, irrelevant, or inapplicable, or if the call obtains no new required information. Refused calls are included. A call containing both a redundant topic and a new question counts as one redundant call. RAR is undefined when the agent makes no calls.

The **Missed Clarification Rate**, MCR_a, pools required clarification slots $\mathcal{U}_\tau^{\mathrm{req}}$ across tasks and measures the fraction still unresolved at task completion. An unresolved slot need not imply that the agent never asked about it. Required slots cover applicable task-specific questions and unresolved preferences needed for reuse. Information already established through retained answers or annotated habit-inference rules does not require another question. Inapplicable branches and slots answered by oracle context in ablations are also excluded.

F LifelongCUAWorld-Revision Protocol Details

F.1 Episode Structure and Feedback Annotations

A Revision episode consists of an initial draft and K_τ feedback rounds. The workspace, project files, and application state persist as the agent revises its deliverable. Each non-final submission saves an artifact snapshot and releases one pre-authored feedback round. The agent receives the initial request and all feedback released so far; the first submission after the last round ends the episode. All agents receive the same feedback sequence.

Feedback is written against the expected artifact from the preceding stage, rather than regenerated from an evaluation of the agent's actual submission. Its references to approved content express the predefined stage requirements. Revision units $\mathcal{R}^{(k)}$ encode feedback requirements, including any explicit constraints to keep content unchanged. Preservation units $\mathcal{P}^{(k)}$ designate protected requirements. The two sets may refer to the same property, but serve different reporting purposes. Each unit is linked to the checkpoints that determine its satisfaction at that stage.

F.2 Stage-Wise Evaluation and Metric Coverage

Evaluation takes place offline on the saved snapshots $\hat{\mathcal{D}}_\tau^{(k)}$. The initial-stage acceptance check determines draft validity and DFR; its outcome does not control feedback release. A revision unit passes when all linked checkpoints pass after revision, whether the requirement was newly achieved or already satisfied. FCR reports this feedback compliance. PS assesses all evaluated protected requirements, while CPR measures retention of the subset $\mathcal{P}^{(k)\circ}$ satisfied in the preceding snapshot (Eq. 2).

Content dimensions have deterministic prerequisites, and a failed prerequisite sets only the dependent dimensions to zero. The final snapshot receives the full artifact evaluation of Section 2.5; FRS averages these scores over all evaluated episodes. FCR, PS, and CPR aggregate units with valid evaluation results from reached feedback rounds. CPR is undefined when no preservation unit was valid before revision. Strict completion requires all stage gates and critical checkpoints to pass.

G Deliverable-Centered Evaluator Details

G.1 Two-Layer Evaluator Configuration

The evaluator uses deterministic rules to check structural and functional validity and content judgments to assess professional quality. Rule checkpoints score observable properties of artifacts or application state as $r_i \in [0,1]$. Content groups combine deterministic checks with rubric judgments based on artifact evidence. Checkpoint weights, critical requirements, rubric dimensions, and score aggregation parameters are fixed before evaluation (Section 2.5). In Expert and Profile, failure of

a source-validity checkpoint gives the dependent content group zero credit. Revision applies these prerequisites to individual rubric dimensions (Appendix F.2).

Each content checkpoint receives three independent judge calls on the same evidence at a fixed temperature. The checkpoint score averages their dimension scores, with a veto only when a majority flags a rubric fail condition. Requirements already covered by deterministic rules are enforced by those rules (Appendix H.3).

G.2 Artifact Evidence Extraction

The judge receives artifact evidence together with the task requirements and rubric (Table 21). Evidence includes extracted text, rendered views, and summaries of native application state. Audio is represented by extracted features, and video by metadata and sampled frames or contact sheets.

Table 21: Artifact evidence for content evaluation, including configured native-state summaries and previews.

Artifact Family	Evidence Representation
PDF and documents	Extracted text, document structure, and rendered pages
Spreadsheets	Sheets, cell values, formulas, metadata, and chart or sheet previews
Images and design files	Raster images, rendered compositions, and available object or layer summaries
Video	Stream metadata, timestamps, and sampled frames or contact sheets
Audio	Text summaries of duration, channels, loudness, peaks, and signal statistics
CAD / 3D / GIS	Geometry, object, layer, or CRS summaries and task-specific rendered views
Code and structured data	Source text, parsed structures, schema checks, and execution outputs

G.3 Workflow Diagnostics

CLI precision, recall, and F1 compare the agent's issued commands with the task's annotated reference commands (Table 5). These diagnostics describe command use and do not contribute to the artifact-based task score.

H Task and Evaluator Validation

H.1 Task Verification

For each task, we check that its initial workspace can be restored, required inputs are available, and the workflow produces inspectable intermediate and final deliverables. We also review whether the evaluator covers the task requirements. Pilot runs check for unstable dependencies and unintended shortcuts. Profile and Revision tasks also undergo the interaction and annotation audits below.

H.2 Evaluator Validation

We test rule checkpoints on reference artifacts and targeted corruptions that violate individual requirements. Content evaluation uses expert-reviewed examples of successful, partially successful, and failed outputs. Fail conditions are checked against the rule layer to avoid duplicate enforcement (Appendix H.3). Repeated and alternative-judge evaluations assess the stability of the resulting scores (Appendix I.5).

H.3 Fail-Condition Audit and Veto Policy

A rubric fail condition sets a checkpoint's score to zero when a majority of the three judge calls flags it. We compare these conditions with deterministic rule checks to avoid enforcing the same requirement twice. Requirements already covered by rules are left to the rule layer. When a condition combines a rule-checkable property with a judgment, only the judgment remains in the rubric. Objective requirements without a corresponding rule check remain in the rubric.

We assess veto-policy sensitivity using saved evidence from the main experiments, with the default results as the reference (Table 22). The stricter alternative retains rule-overlapping fail conditions,

uses two judge calls, and vetoes a checkpoint if either call flags a failure. Among 160 rule-passing content checkpoints, six verdicts differ between the policies. Re-aggregated suite metrics change by less than 0.005, with rankings unchanged among the systems included in this comparison.

Table 22: Sensitivity to the veto policy on a fixed set of trajectory evidence. Strict: rule-overlapping fail conditions, two calls, veto on either flag; Default: audited conditions, three calls, majority veto. Judged: rule-passing content checkpoints; Vetoed: checkpoints receiving zero by veto; Passed: checkpoints meeting the final pass criteria after deterministic prerequisite checks.

		Checkpoints (strict→default)			**Suite metric**	
Suite	**Model**	Judged	Vetoed	Passed	Strict	Default
Expert (Score)	Qwen3.7-Plus	9	3→1	3→5	0.053	0.058
	GLM5.3-flash	18	4→1	7→7	0.113	0.114
	Kimi K3	4	1→1	3→3	0.028	0.028
	MiniMax M3	7	0→0	5→4	0.032	0.030
Profile (PTS)	GPT-5.4	3	1→1	0→0	0.044	0.044
	Qwen3.7-Plus	4	2→2	0→0	0.047	0.047
	GLM5.3-flash	7	1→0	3→4	0.100	0.104
	Kimi K3	3	0→0	1→1	0.051	0.051
	MiniMax M3	8	3→3	1→1	0.054	0.054
Revision (FRS)	Qwen3.7-Plus	33	17→10	0→0	0.021	0.021
	GLM5.3-flash	52	22→19	1→1	0.066	0.066
	Kimi K3	1	1→1	0→0	0.000	0.000
	MiniMax M3	11	4→4	0→0	0.008	0.008
All		160	59→43	24→26		

H.4 Profile and Revision Consistency Audits

For Profile, we check that histories, canonical answers, answer availability, and preference checkpoints are consistent with the hidden profile. Cross-task checks verify that preferences can be reused as intended and that current instructions and contextual exceptions take priority. For Revision, we check initial requirements, feedback release, snapshot integrity, and the checkpoints linked to each unit. The saved evidence must support the specified feedback and preservation checks, including their different eligibility conditions.

I Supplementary Experiments

I.1 Model Settings

Table 23 reports the decoding and context settings used in the main experiments. All main-experiment trajectories follow the default configurations in Section 3.1. DeepSeek Harness and all model baselines use a step limit of twice the estimated step count. Thinking labels specify the reasoning mode or effort, and dashes denote unspecified sampling parameters. History is the number of recent screenshots retained (MiniMax M3 also retains the first); Native denotes SDK-managed context. Ablations retain these model settings and vary the factors specified in Section 4. Online `ask_user` is disabled in the Profile information ablation.

Table 23: Model decoding settings for agent action generation.

System	**Thinking**	**Temp.**	**Top-p**	**History**
Agent Harness				
DeepSeek Harness + DeepSeek-V4.1-Flash	High	—	—	Native
Proprietary models				
GPT-5.4	high	1.0	—	3
GPT-5.6 Sol	Medium	1.0	—	3
GPT-5.6 Luna	Extra high	1.0	—	3
Seed-2.1-Pro	Thinking	1.0	0.7	5
Qwen3.7-Plus	High	0.0	0.9	20
Open-weight models				
DeepSeek-V4.1-Flash	High	—	—	3
GLM5.3-flash	High	1.0	0.95	4
Kimi K3	High	1.0	—	3
MiniMax M3	Thinking	0.6	—	10

I.2 Expert Interface and CLI-Knowledge Ablation

The Expert ablation varies interfaces and CLI knowledge for GLM5.3-flash. Table 5 specifies the six conditions, separating interface access from the scope and specificity of command guidance. Section 4.1 compares scores and execution diagnostics.

I.3 User Information and Preference Application

With online `ask_user` disabled, the Profile ablation varies supplied context: profile disclosure, history, and current or prior oracle Q/A (Table 6). Adding prior answers while holding current-task answers fixed tests cross-task information dependencies; it does not test autonomous acquisition of those answers. Results appear in Section 4.2.

I.4 Revision Protocol and State-Persistence Ablation

The Revision ablation varies draft source, feedback delivery, or workspace persistence while fixing the episodes, budget, and final evaluation requirements (Table 7). The default uses agent drafts, staged feedback, and workspace carryover.

Episodes and conditions. The subset contains 18 episodes, two per scenario, each with unit annotations and a human-authored canonical draft $D^{(0)*}$. Default results reuse the corresponding main runs. The oracle-draft condition starts from $D^{(0)*}$ at the first feedback round. Full instruction presents the request and all feedback together; without intermediate submissions, draft and unit-level metrics are undefined. W1 retains only preceding deliverables; W2 retains none. Both receive the request, released feedback, and stage assets.

Interpreting the comparisons. The oracle-draft condition improves FCR, FRS, PS, and CPR; removing workspace carryover lowers CPR with little FCR change (Section 4.3). The unit-level rates retain their eligibility conditions (Appendix F.2), supporting complementary diagnostics rather than independent measures of modification and preservation ability.

I.5 Stability of the Content-Layer Evaluation

We assess content-judge stability on 160 saved checkpoints from thirteen agent–suite pairs: 38 Expert, 25 Profile, and 97 Revision. These additional evaluations only assess evaluator stability. We repeat evaluation with Gemini-3.1-Pro and substitute six alternative judges, keeping rubrics, prompts, temperature, and the three-call protocol fixed. Both sides of Table 24 use averaged and weighted dimension scores, majority veto, and total-score and critical-dimension thresholds, without applying deterministic prerequisites. Table 25 re-aggregates suite scores using these judgments and the original rule scores.

Table 24: Content-judge agreement on saved checkpoints. All judges evaluate the same 160 checkpoints with three valid judgments each. The recorded pass rate is 25.0% (40/160). Agree: agreement with recorded judge verdicts; $|\Delta|$: mean absolute score difference; κ: Cohen's κ; Pass: pass rate.

Judge	Agree (%)↑	$\lvert\Delta\rvert$↓	κ↑	Pass (%)
Gemini-3.1-Pro (re-run)	95.6	0.067	0.880	23.1
Gemini 3.6 Flash	75.6	0.146	0.497	46.9
Gemini 3.1 Flash-Lite	46.9	0.336	0.146	74.4
Qwen3.5-397B-A17B	71.3	0.205	0.418	48.8
Qwen3.6-Plus	63.8	0.258	0.333	58.8
Qwen3.6-27B	61.9	0.260	0.326	63.1
Qwen3.5-35B-A3B	65.0	0.248	0.317	52.5

Repeated evaluation with the same judge is stable. Repeated Gemini-3.1-Pro evaluation agrees with the recorded judge verdicts on 95.6% of checkpoints, with a mean absolute score difference of 0.067. Cohen's κ is 0.880. After aggregation, suite scores change by less than 0.005 and model rankings remain unchanged within the evaluated combinations (Table 25).

Judge choice affects checkpoint decisions more than aggregate scores. Alternative judges agree with the recorded verdicts on 46.9–75.6% of checkpoints and assign higher pass rates (46.9–74.4%, compared with the recorded 25.0%). After re-aggregation, Expert and Profile scores change by up to approximately 0.02 (Table 25). Revision FRS is unchanged for three of the four evaluated systems and differs by at most 0.003 for GLM5.3-flash. Within this comparison, the highest-scoring system in each suite remains the same across judges, although some closely scored systems change order. These results distinguish repeatability with a fixed judge from sensitivity to the choice of judge.

Table 25: Stability of the suite metrics under different judges. Expert Score and Profile PTS of the main-table models when every rule-passing content checkpoint is scored by the given judge and the suite metric is re-aggregated with the rule layer unchanged. τ: Kendall's τ between the resulting model ranking and the recorded one. Revision FRS is omitted: it is identical for Kimi K3, MiniMax M3 and Qwen3.7-Plus under every judge and within 0.003 for GLM5.3-flash.

	Expert Score					**Profile PTS**					
Judge	Qwen3.7	GLM5.3	Kimi	MiniMax	τ	GPT-5.4	Qwen3.7	GLM5.3	Kimi	MiniMax	τ
Recorded (Gemini-3.1-Pro)	0.058	0.114	0.028	0.030	–	0.044	0.047	0.104	0.051	0.054	–
Gemini-3.1-Pro (re-run)	0.055	0.114	0.028	0.030	1.00	0.044	0.047	0.100	0.051	0.054	1.00
Gemini 3.6 Flash	0.062	0.127	0.028	0.036	1.00	0.051	0.062	0.104	0.058	0.070	0.80
Gemini 3.1 Flash-Lite	0.061	0.131	0.028	0.032	1.00	0.066	0.062	0.116	0.066	0.070	0.80
Qwen3.5-397B-A17B	0.057	0.127	0.028	0.034	1.00	0.051	0.055	0.104	0.058	0.070	1.00
Qwen3.6-Plus	0.059	0.125	0.035	0.034	0.67	0.058	0.054	0.104	0.051	0.075	0.40
Qwen3.6-27B	0.063	0.135	0.035	0.036	1.00	0.051	0.062	0.104	0.058	0.075	0.80
Qwen3.5-35B-A3B	0.060	0.129	0.028	0.034	1.00	0.066	0.062	0.104	0.066	0.074	0.80

J CASE STUDIES

These cases show how professional requirements are checked in submitted deliverables throughout task execution and user collaboration. Each case links requirements, agent actions, artifact evidence, and evaluation results to explain successes and incomplete work.

J.1 EXPERT: FROM DELIVERABLES TO EVALUATION

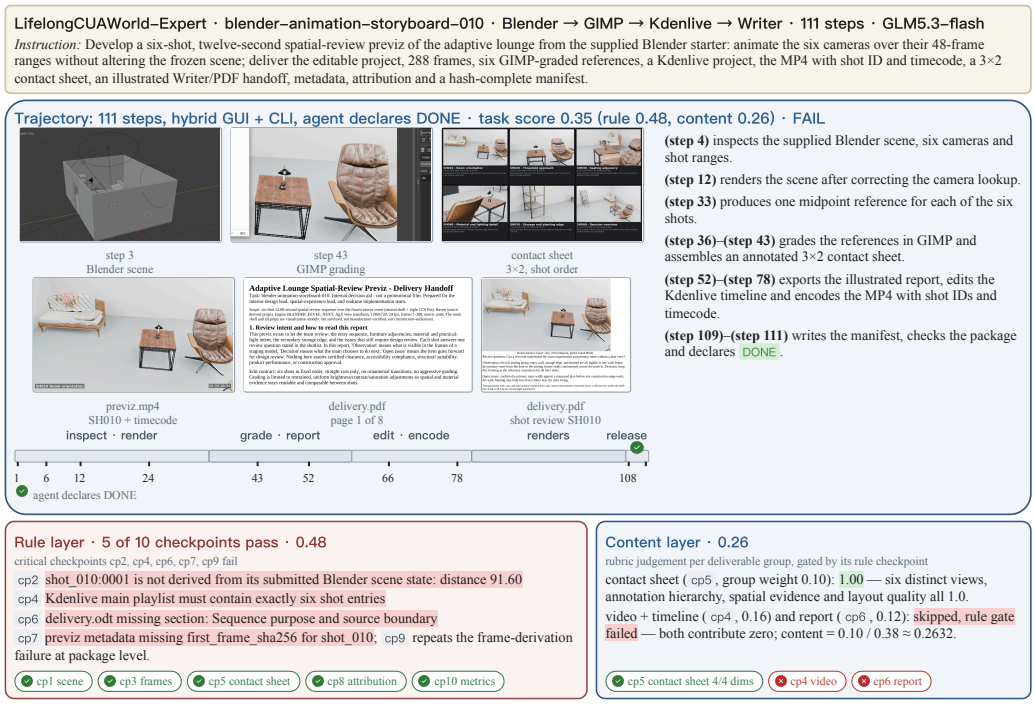


Figure 6: Expert case study: evaluating an animation delivery package. The figure follows GLM5.3-flash from execution across multiple applications to submitted artifacts and checkpoint verdicts. In the recorded evaluation, the contact sheet receives full content credit, while failed rule checks block content credit for the video and report. Content scores are aggregated using group weights normalized by their sum. The task scores 0.35 and fails strict completion.

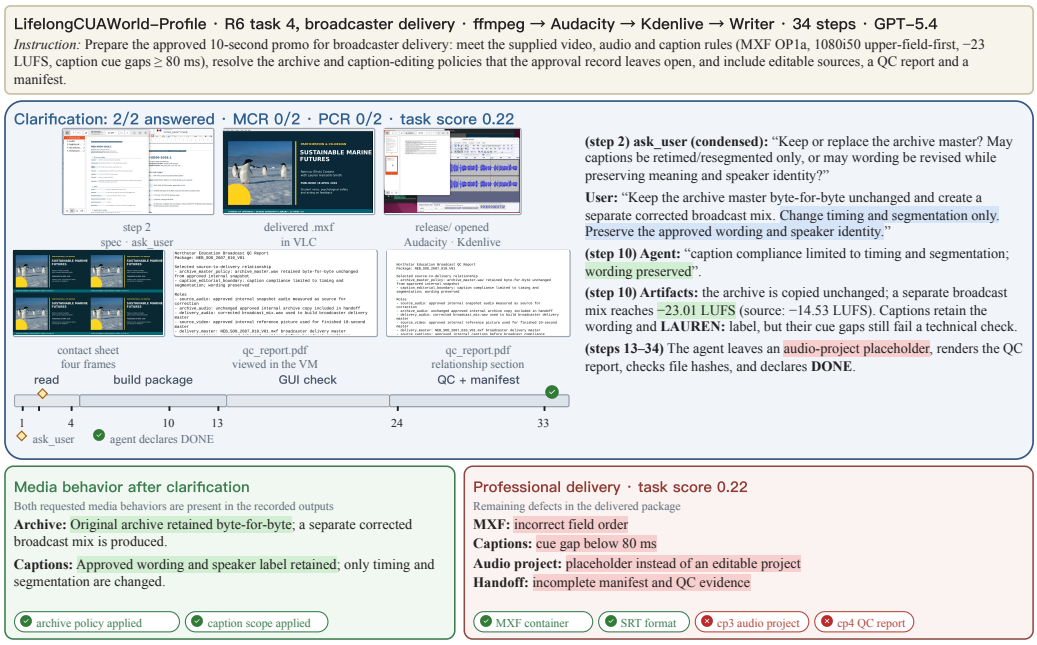
The spatial-review task tests whether work across several applications forms a consistent, editable delivery package (Figure 6). GLM5.3-flash must produce a six-shot, twelve-second animation together with its Blender project, rendered frames, contact sheet, and handoff documents. It uses Blender scripts for rendering, the GIMP CLI for color adjustment, and Kdenlive for video assembly, then prepares the report and delivery manifest. Evaluation checks the artifacts against the same shot sequence and delivery requirements.

From artifact checks to the task verdict. The deliverables pass five of ten rule checkpoints, with failures in frame provenance, timeline structure, and handoff documentation. The contact sheet receives full credit on all four content dimensions; the video and report receive zero after failing their rule gates. The group weights for video, contact sheet, and report are 0.16, 0.10, and 0.12, giving a normalized content score of $0.10/(0.16 + 0.10 + 0.12) \approx 0.2632$. Combining this with the rule score of 0.48 yields a task score of approximately 0.350. This is below the 0.80 threshold, and failed critical checks also prevent strict completion.

J.2 Profile: Preference Application and Artifact Quality

Figure 7 follows GPT-5.4 through clarification and broadcast delivery. The agent asks about archive retention and caption editing. The user requests an unchanged archive with a separate broadcast mix, and permits caption retiming and resegmentation while preserving wording and speaker identity. Both required clarification slots are answered, with no redundant questions (MCR 0/2; RAR 0/1).

Both requested media behaviors are present in the recorded outputs. The archive remains unchanged, and the captions retain the approved wording and speaker label. The package nevertheless has incorrect MXF field order, insufficient caption gaps and a placeholder audio project. These defects leave the overall task score at 0.22. PCR is 0/2 because the full preference-related units also require editable-project and structured QC evidence. The case distinguishes obtaining user information and applying media preferences from satisfying the complete delivery requirements assessed by PCR.



Recorded PCR: 0/2: media behaviors are applied, but the required project/QC evidence is incomplete. Both required questions answered: MCR 0/2, RAR 0/1.

Figure 7: Profile case study: preference application and artifact quality. GPT-5.4 obtains answers to both required questions (MCR 0/2) and preserves the archive and caption text. The delivery package remains incomplete (task score 0.22). Its missing editable-project and QC evidence leave the full preference-related units unsatisfied (PCR 0/2).

J.3 Revision: Feedback Compliance and Preservation

These cases examine feedback satisfaction and content preservation in professional artifacts. Figures 8–10 follow GLM5.3-flash in risk assessment, map production, and gallery design, complementing the music-education case in Figure 5. Feedback is pre-authored against expected stage artifacts (Appendix F.1). The saved snapshots show which requirements hold after revision and whether previously correct content remains intact.

Regression after a valid draft (Figure 8). This case tests whether an agent can extend a risk assessment while retaining an approved baseline decision. The draft passes the initial-stage check. The first feedback requests six additional risks and twelve controls for two approved integration paths while preserving the baseline assessment. The agent adds material, but its risk scores and assigned owners conflict with the requirements, so all three revision units fail. It also omits the approved internal-pilot decision from the memo. The second revision completes the verification matrix and correctly concludes NO-GO after the tenant-isolation test fails. Across both rounds, FCR is 1/6, PS is 3/6, and CPR is 3/4. Comparing the snapshots separates the loss of the approved decision from errors in the requested additions.

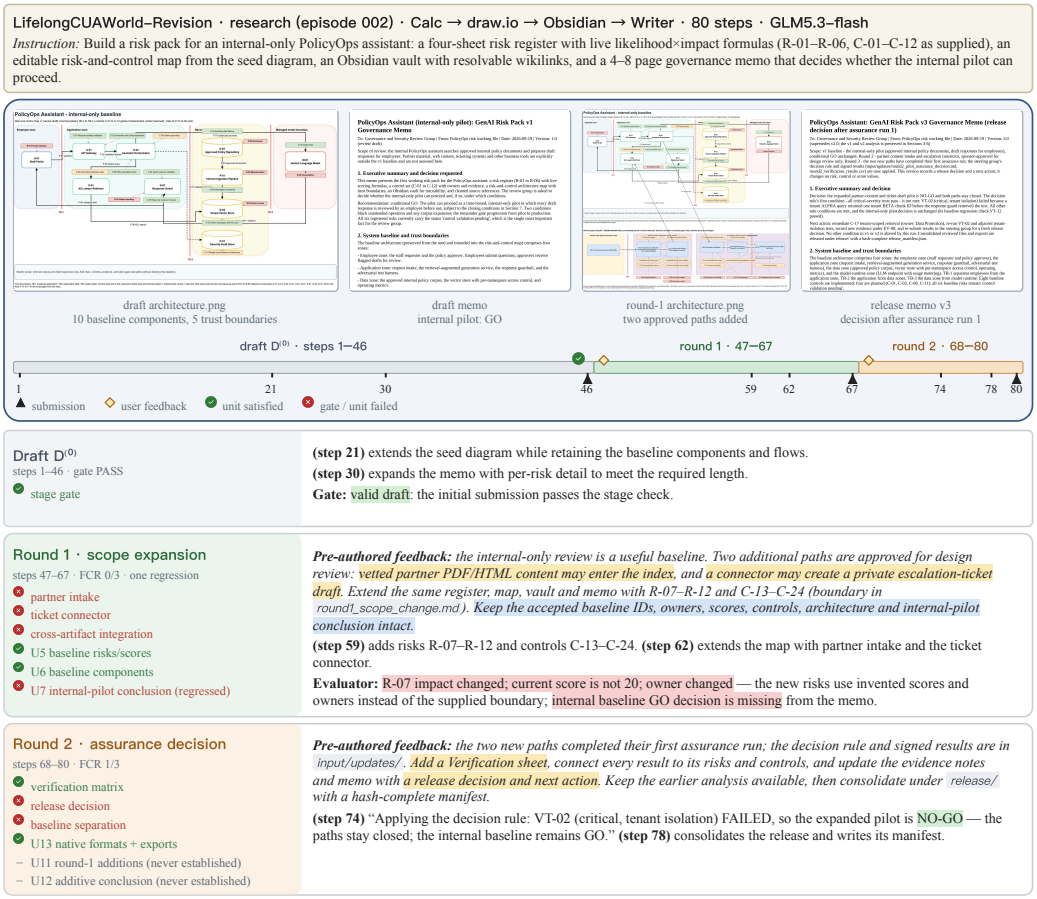


Figure 8: Revision case study: extending a risk assessment while preserving its baseline. The stage snapshots and unit verdicts distinguish errors in newly added risks from loss of an approved decision. GLM5.3-flash omits the internal-pilot decision in its first revision and completes the verification matrix in the second. Across both rounds, FCR is 1/6 and CPR is 3/4.

Preserving protected files during revision (Figure 9). The permit-atlas case tests whether an agent can produce new maps while keeping designated protected files unchanged. Before each revision, the agent records file checksums. In the second round, these checks reveal unintended changes to GeoPackage files opened in QGIS; the agent restores the originals before submission. All four previously valid preservation units are retained (CPR 4/4). The maps still omit a style category and area totals and use the wrong atlas order. Only the final screen edition passes its revision check (FCR 1/6). File preservation and map content are checked separately, so successful recovery does not obscure the unfinished revisions.

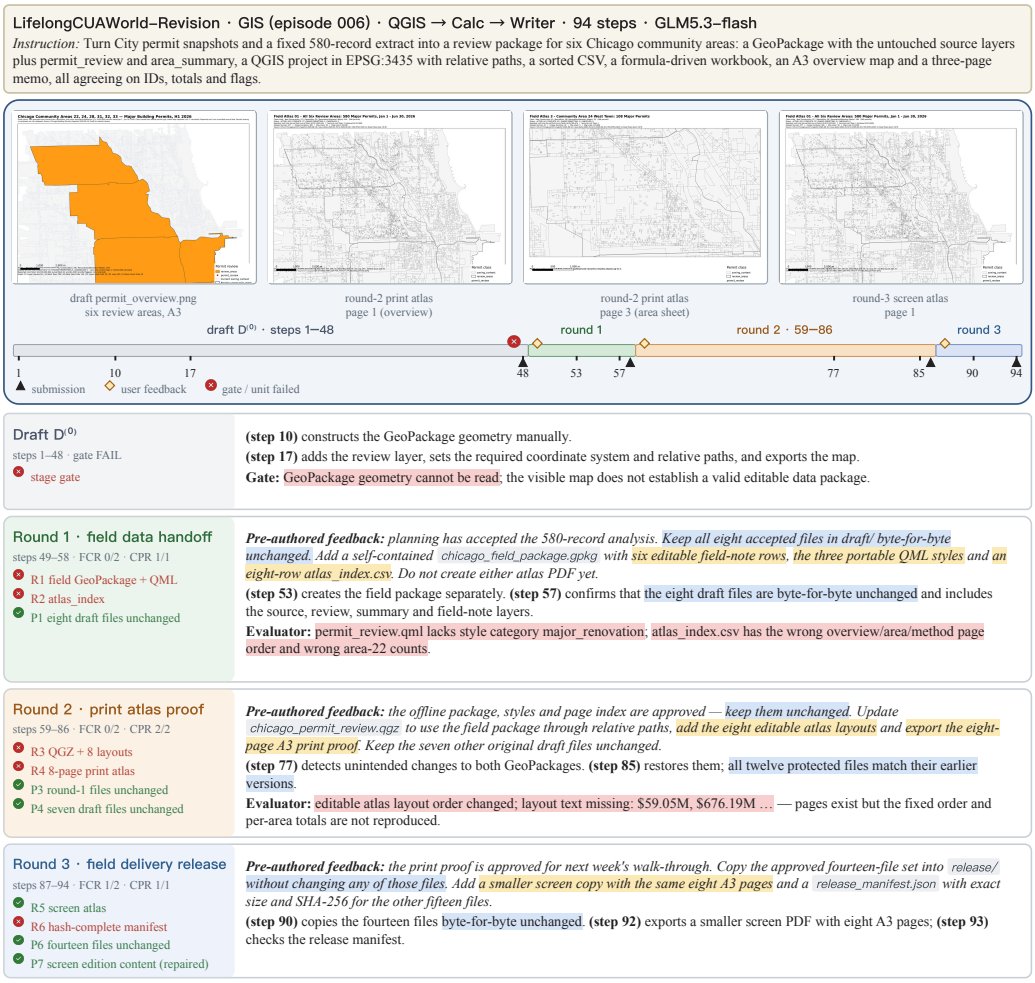


Figure 9: Revision case study: file preservation and map quality in a permit atlas. The feedback specifies the expected stage requirements in advance. GLM5.3-flash detects unintended file changes and restores the designated GeoPackages before submission (CPR 4/4). Feedback checks identify missing style and area information and incorrect atlas ordering; only the screen edition passes (FCR 1/6).

Unintended changes to protected objects (Figure 10). The gallery case tests whether an agent can apply revised site measurements while preserving an approved exhibit arrangement. The requested changes concern partitions, lighting, and visitor routes; source building objects, cameras, and materials must remain intact. The agent preserves these elements but moves exhibit G-11 while lowering the track lights. Route and headroom requirements remain unmet. Its only passing feedback unit is camera retention (R3), a property also covered by preservation unit P3. Thus, FCR 1/4 records a satisfied feedback constraint, not a newly completed modification. CPR 2/3 records retention among previously satisfied protection requirements; the displaced exhibit fails its preservation check.

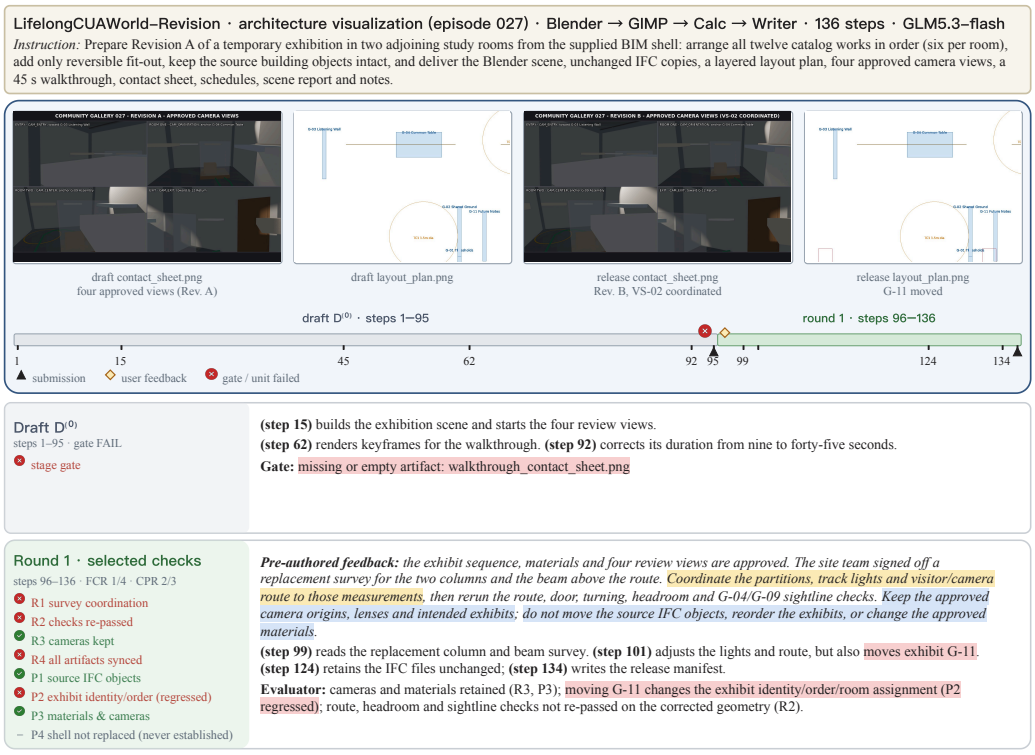


Figure 10: Revision case study: revising a site while preserving approved exhibits. GLM5.3-flash displaces exhibit G-11 while revising lighting. Camera retention satisfies feedback unit R3 and also appears in preservation unit P3. Selected checks are shown; scores cover all evaluated units (FCR 1/4, PS 3/5, CPR 2/3).