Computer-Use Agent Selection Checklist

Fora Soft Learn · AI for Video Engineering

1 - What counts as a computer-use agent?

- [ ] It operates software like a person - looks at the screen, clicks, types - with no custom back-door built in advance.
- [ ] It runs a loop: OBSERVE (screenshot) -> REASON (a vision-language model picks one action) -> ACT (click/type) -> observe again.
- [ ] If the target system already has a proper interface for software to use, a scoped integration is faster, cheaper, more reliable. Use the agent only when there is no back-door.

2 - Which question are you asking? (Track A vs Track B)

```
Track B - build faster : agent operates YOUR dev tools (code, terminal). Mistake = bad commit, caught in review.
Track A - act for user : agent operates the product FOR the user. Mistake = wrong action on a real account.
Two separate decisions, two different risk profiles. Do not conflate them.
```

3 - The five 2026 products at a glance

```
Claude Code      (B) : writes/tests/commits code from plain language. Best at migrations, refactors, CI fixes.
OpenAI Operator  (A) : drives a web browser; now ChatGPT 'agent mode'. Standalone site retired Jul 2025.
Manus AI         (A) : long autonomous jobs from one prompt. Credit pricing = unpredictable; owned by Meta (Dec 2025).
Perplexity Comet (A) : Chromium AI browser; does browsing busywork. An agent BESIDE your app, not inside it.
Apple Intelligence(A): on-device, privacy-first; Siri + App Intents expose SCOPED actions. Apple platforms only.
```

4 - The OSWorld reality check (set expectations)

- [ ] Measure agents against the ~72% HUMAN baseline on OSWorld (369 real desktop tasks), not against 100%.
- [ ] Web-only tasks are far easier (~87% WebVoyager) than full desktop control (Operator CUA scored 38.1% on OSWorld).
- [ ] Reliability compounds: five steps at 90% each = 0.9^5 = ~59% end-to-end. Long chains fail more than the demo suggests.

5 - Build, buy, or skip - and the three mandatory guardrails

- [ ] Track B: use a coding agent (Claude Code / Cursor). Keep tests green, review every commit, own the architecture.
- [ ] Track A: if the task is web-shaped and narrow, an agent is production-ready. If it needs broad desktop control, wait or constrain.
- [ ] GUARDRAIL 1 - human approval on any action that spends money, sends a message, or deletes data.
- [ ] GUARDRAIL 2 - a hard cap on cycles per task (so a confused agent cannot loop forever).
- [ ] GUARDRAIL 3 - a hard cap on spend (each observe-reason-act cycle is billed; ~1-2 cents adds up fast on long jobs).