Neuphlo

TECHNICAL WHITEPAPER

Agents that do the work — not just chat about it.

Inside Neuphlo's agent platform: @mentionable AI teammates with their own persona, tools and memory, an orchestrator that delegates across specialists, and EU AI Act governance built into every agent.

Autonomous execution · Multi-agent · EU AI Act ready

neuphlo.com · 2026

01 — FROM COPILOTS TO COLLEAGUES

§ Why agents, not just a chatbot

Most AI features bolted onto productivity tools are copilots: a side panel that drafts text when you ask. Useful, but passive. Neuphlo takes a different stance — agents are **members of the workspace** you can @mention, hand a task to, and trust to act.

An agent in Neuphlo isn't a prompt box. It's a named teammate with a personality, a defined job, a scoped set of tools it's allowed to use, and a memory of what it has done before. You bring one into a conversation the same way you'd loop in a colleague — `@sprint-planner, can you break this down?` — and it actually does the work: reads the relevant tasks, searches the knowledge base, creates sub-tasks, posts a plan, and tells you what still needs a human.

@mention	100+	18
to invoke any agent	tools agents can use	ready-made specialists

What makes them different

- **They take action.** Agents call real tools — create and update tasks, write knowledge-base articles, search the workspace, send messages, draft and publish posts — not just produce text.
- **They collaborate.** A built-in orchestrator routes work to the right specialist and brings results back, so one mention can mobilise a whole team of agents.
- **They're observable.** Every run is a visible, step-by-step timeline; every agent has a dashboard of its activity, success rate and token use.
- **They're governed.** Each agent carries an EU AI Act risk classification, an optional human-oversight kill switch, and an always-on AI disclosure.

> **This document** walks through the anatomy of a Neuphlo agent, how the execution loop turns an instruction into real work, how agents delegate to one another, how you observe and govern them, and how the whole system stays accountable under EU law. The screenshots are taken directly from the product.

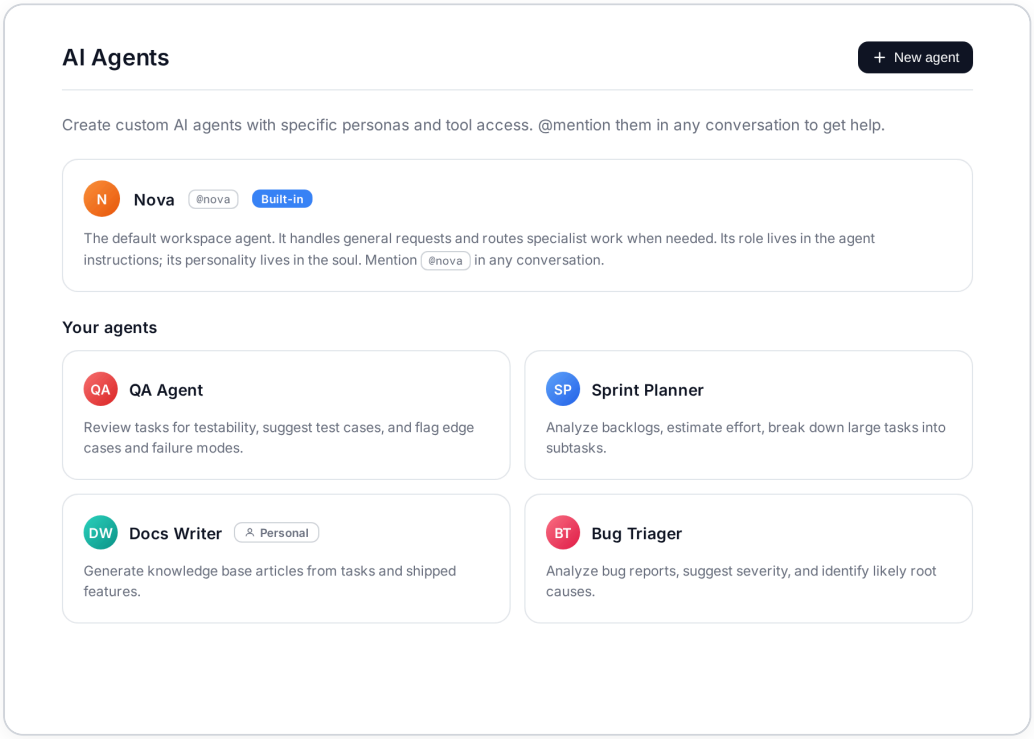
02 — ANATOMY OF AN AGENT

§ What an agent is made of

Every agent is built from four parts. Together they define who it is, what it does, and what it's allowed to touch.

PART	WHAT IT CONTROLS
Soul	The personality — tone, qualities, communication style. A QA agent is skeptical and precise; an onboarding guide is warm and patient.
Instructions	The job — the step-by-step role the agent performs when invoked ("review the task, generate test cases, flag issues...").
Tool scope	The permissions — exactly which of the 100+ workspace tools this agent may call, and which are required. Tools outside the scope simply don't exist for it.
Model	The engine — the default workspace model, or a specific provider/model per agent (including local models when self-hosted).

Splitting persona ("soul") from role ("instructions") is deliberate: two agents can share a job but differ in temperament. Tool scope is the safety boundary — an agent meant only to read and summarise is never granted the tools that create or delete.



Cowork › AI Agents. The built-in orchestrator (Nova) sits above your custom specialists; each agent is invoked by @mention anywhere in the workspace.

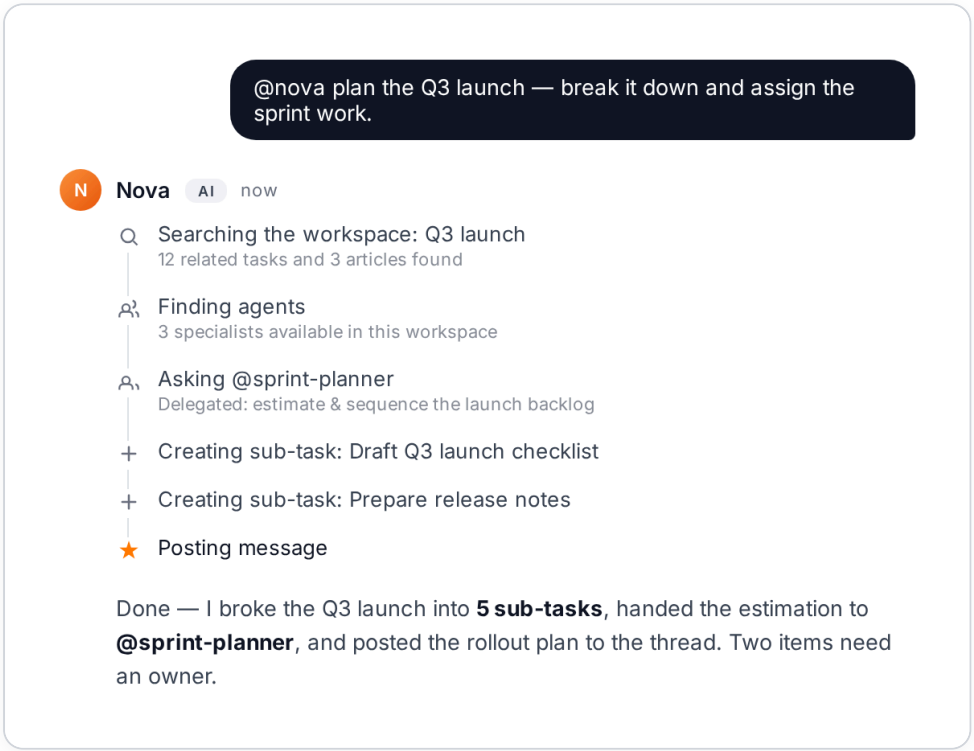
You don't have to start from scratch: ship from the library, clone and tune one, build from a blank slate, or simply **describe what you need** and have Nova draft the agent for you.

§ The execution loop

When an agent is invoked, it doesn't answer in one shot. It runs an agentic loop: think, call a tool, read the result, decide the next step — repeating until the work is done.

Each iteration the model picks a tool, Neuphlo executes it against the real workspace on the user's behalf, and the result feeds the next decision. The loop runs up to **20 tool iterations** per turn, enough to chain a meaningful piece of work — search, then read, then create, then post — while staying bounded. Read-only tools (searching, listing, reading) run freely; tools that change data are treated with more care, and high-risk agents confirm every mutation.

The whole run streams to the conversation as it happens, so the agent's reasoning is never a black box:



A live agent run. Each step is a real tool call — searching the workspace, delegating to a specialist, creating sub-tasks, posting a result — shown as it executes, then preserved with the agent's reply.

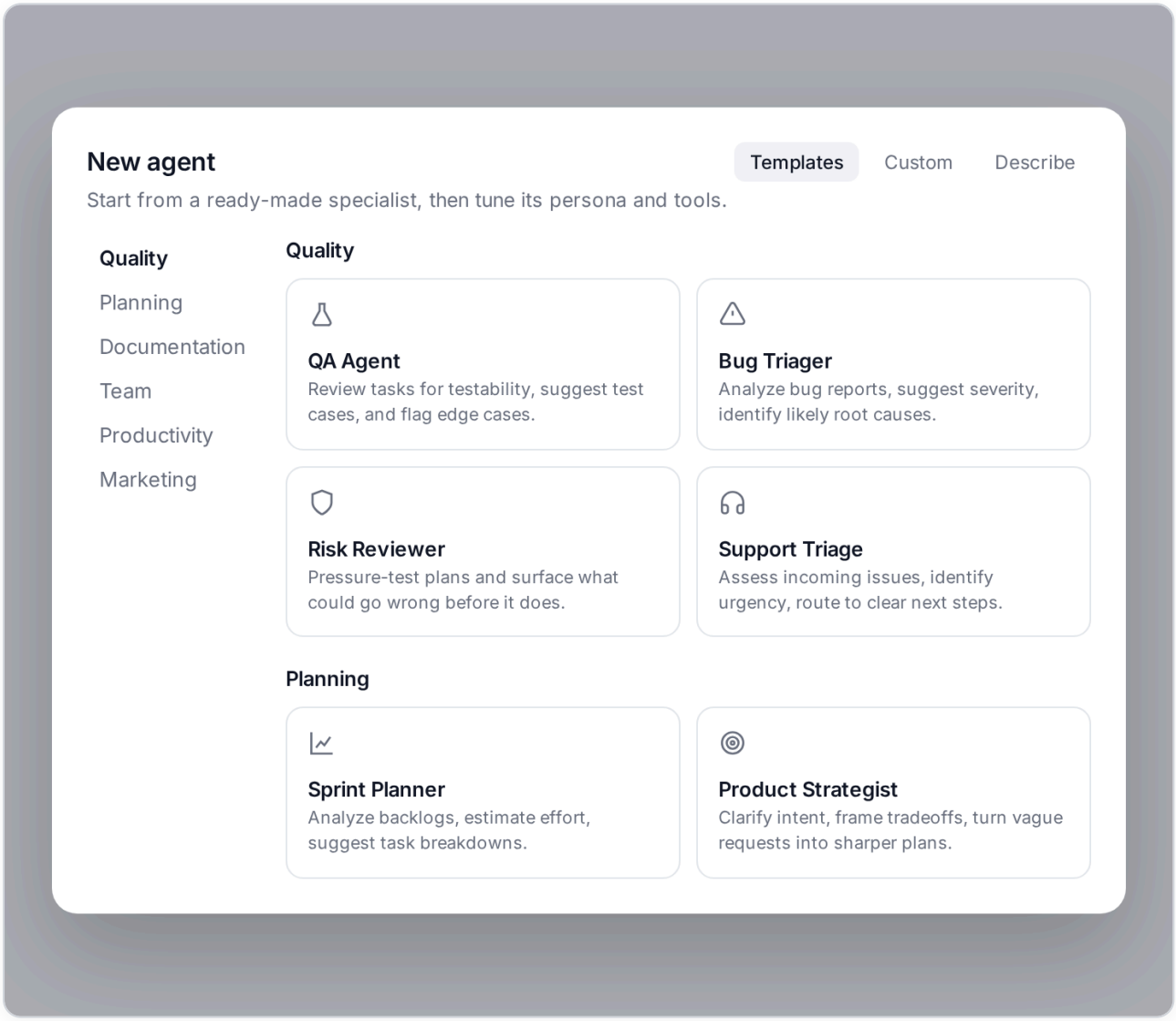
Two more things happen quietly around the loop. Before it starts, the agent retrieves relevant **memories** from past runs so it improves over time; and a run can be **cancelled mid-flight**, stopping the agent between tool calls.

04 — A TEAM, NOT A TOOL

§ Specialists & orchestration

The real power isn't one clever agent — it's many narrow ones that hand work to each other. Neuphlo's default agent, **Nova**, is an orchestrator: it fields general requests and delegates specialist work to the right agent.

Nova can discover the other agents in a workspace and call them in two ways. For a quick answer it *asks* a specialist inline and lets them reply directly in the thread; for durable, tracked work it *assigns a task* to the specialist, which then runs its own loop. The handoff reads like a group chat — Nova tags the specialist in, the specialist answers, and Nova only speaks again when there's a real next step.



The starter library. Eighteen specialists across Quality, Planning, Documentation, Team, Productivity and Marketing — each pre-wired with a persona and a sensible tool scope you can adjust.

Why narrow agents beat one generalist

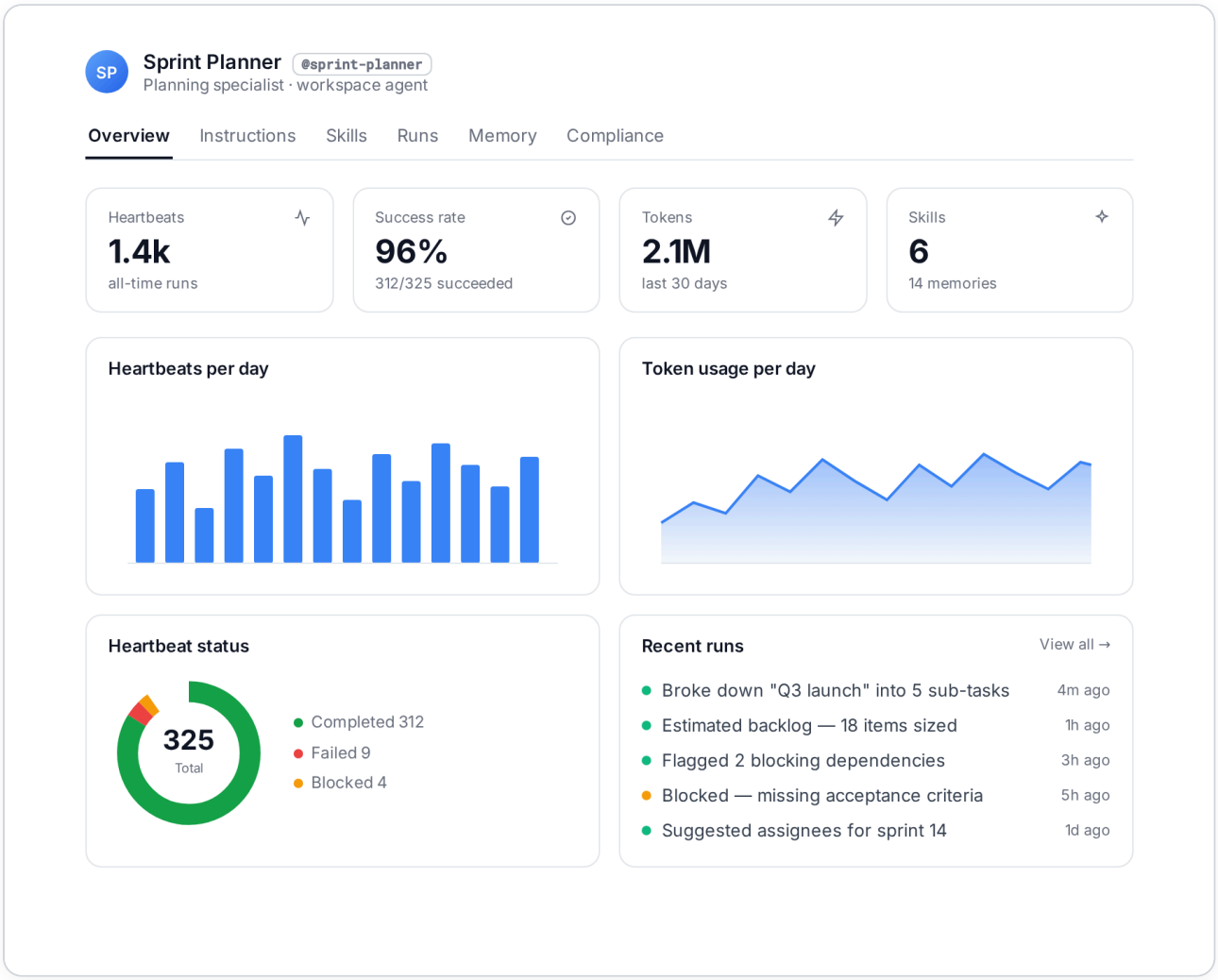
A focused agent has a tighter prompt, a smaller tool scope, and a clearer success criterion — which makes it more reliable and easier to govern than a single do-everything bot. Composition does the rest: a launch request can fan out to a planner, a docs writer and a release-notes writer, each doing the part it's best at, coordinated by Nova. Specialists range from a QA reviewer and

05 — PROACTIVE & ACCOUNTABLE

§ Autonomy you can see

Agents aren't only reactive. They can be assigned tasks they then work autonomously, run on a schedule, and react to events — and everything they do is measured.

Assign a task to an agent and it picks it up, plans the work, and reports structured progress. A daily-briefing automation makes the orchestrator feel proactive rather than purely on-demand. And because autonomy without visibility is a risk, every agent has its own dashboard:



Per-agent overview. Heartbeats (runs), success rate, token usage, skills and memories — plus daily activity charts and a live feed of recent runs with their outcomes.

Heartbeats & success rate

Every run is recorded with its outcome — completed, blocked or failed — so you can spot a misbehaving agent at a glance instead of discovering it in the work.

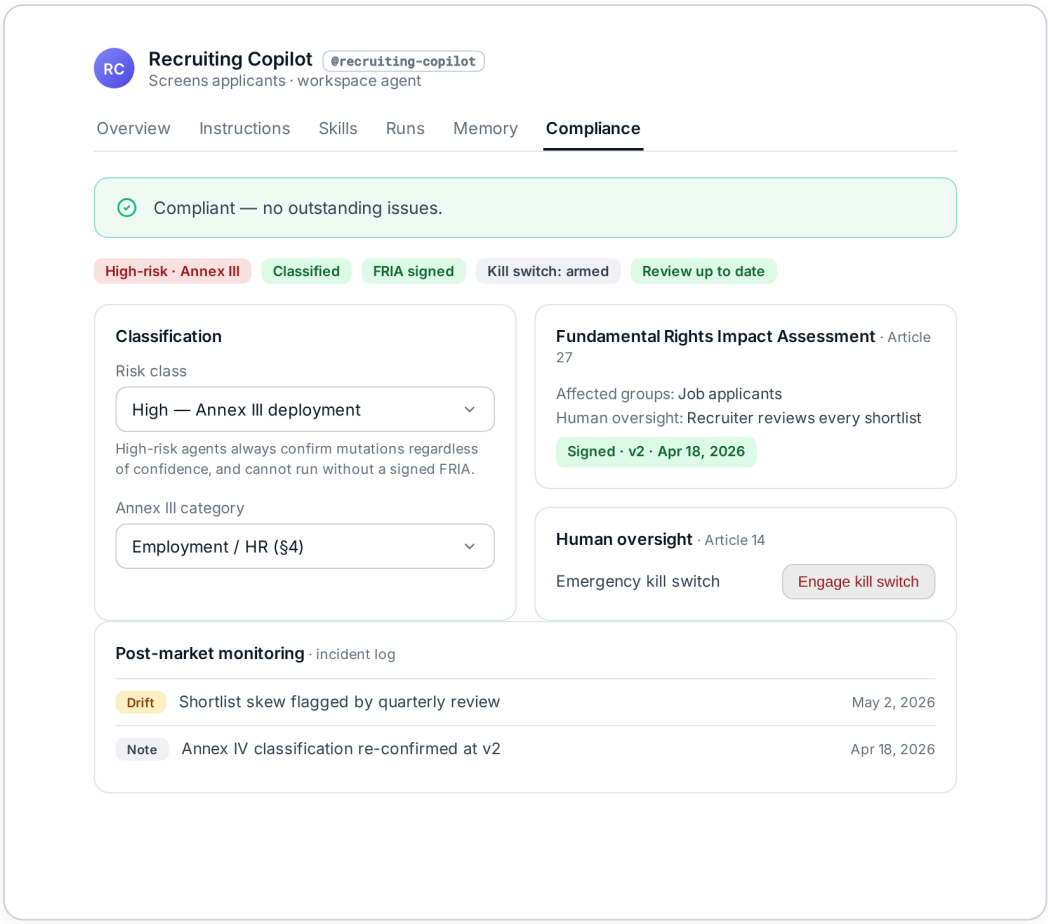
Cost in the open

Token usage is tracked per agent and per day, so the cost of autonomy is always visible rather than buried in an aggregate bill.

06 — BUILT FOR EU REGULATION

§ Governance & the EU AI Act

Autonomous agents acting on real data raise a real question: who is accountable, and how? Neuphlo answers it in the product. Every agent carries a first-class compliance surface mapped to the EU AI Act.



The compliance tab. Risk classification, the Article 27 impact assessment, the Article 14 kill switch, classification history and a post-market incident log — per agent.

Each agent is classified by risk — **minimal**, **limited**, or **high** (an Annex III deployment). The classification isn't cosmetic: a high-risk agent always confirms data changes regardless of confidence, and *cannot run at all* without a signed Fundamental Rights Impact Assessment.

- **Article 27 — FRIA.** High-risk agents require a Fundamental Rights Impact Assessment recording affected groups, identified risks, human-oversight measures and mitigations.
- **Article 14 — human oversight.** A per-agent kill switch lets an owner immediately stop an agent from running.
- **Article 50 — transparency.** Agent messages always carry an "AI" disclosure, and users are explicitly informed they're interacting with an AI system.
- **Annex IV & post-market monitoring.** Classification changes are versioned, and an incident log captures drift and issues over time.

§ Trust by architecture

Governance on top only matters if the foundation is trustworthy. Neuphlo's agents inherit the platform's privacy-first posture: tool calls run with scoped permissions on behalf of a real user, agent activity is fully auditable, and — on a self-hosted deployment — agents can run against **local models** so that prompts, documents and customer data never leave your infrastructure.

- **Scoped by default.** An agent can only call the tools in its scope, acting within the permissions of the workspace — never more.
- **Auditable.** Runs, classifications and incidents are recorded, giving a paper trail for both debugging and compliance.
- **Yours to host.** Self-host Neuphlo and pair local model providers, and the entire agent loop — including the AI — stays inside your own network.

§ Getting started

Bringing an agent into your workspace takes minutes, by whichever route fits:

ROUTE	BEST FOR
Pick a template	Start from one of 18 specialists and adjust the persona and tools.
Describe it	Tell Nova what you need in plain language; it drafts the agent for you.
Build custom	Write the soul, instructions and tool scope yourself for full control.

Then @mention it in any conversation, or assign it a task — and review its work from the run timeline and dashboard.

> **In short:** Neuphlo agents are governed, observable teammates that take real action and collaborate with one another — autonomy you can actually trust, built for teams that have to answer for how AI is used.

Neuphlo

Learn more & try it → **neuphlo.com**