

USE CASE · SOFTWARE & PRODUCT TEAMS

# Ship more. Coordinate less.

How software and product teams run sprints, bugs and launches in Neuphlo — with AI agents picking up the planning, triage and QA work that usually eats the team's week.

## § The coordination tax

Most engineering time isn't lost writing code — it's lost around it: breaking down vague requests, triaging bugs, chasing statuses, writing the test cases nobody had time for.

Teams stitch this together across four or five tools: a tracker here, chat there, docs somewhere else, and an AI assistant in a separate tab that can see none of it. Every boundary between those tools is a place where context dies — and where a human has to ferry it across by hand.

Neuphlo's bet is different: put tasks, conversations, knowledge and testing in **one workspace**, then let agents work across all of it with the same tools a teammate would use. The result isn't "AI features" sprinkled on a tracker — it's a tracker where the routine work starts doing itself.

1

workspace for tasks, chat, docs  
& tests

@mention

to put an agent on the job

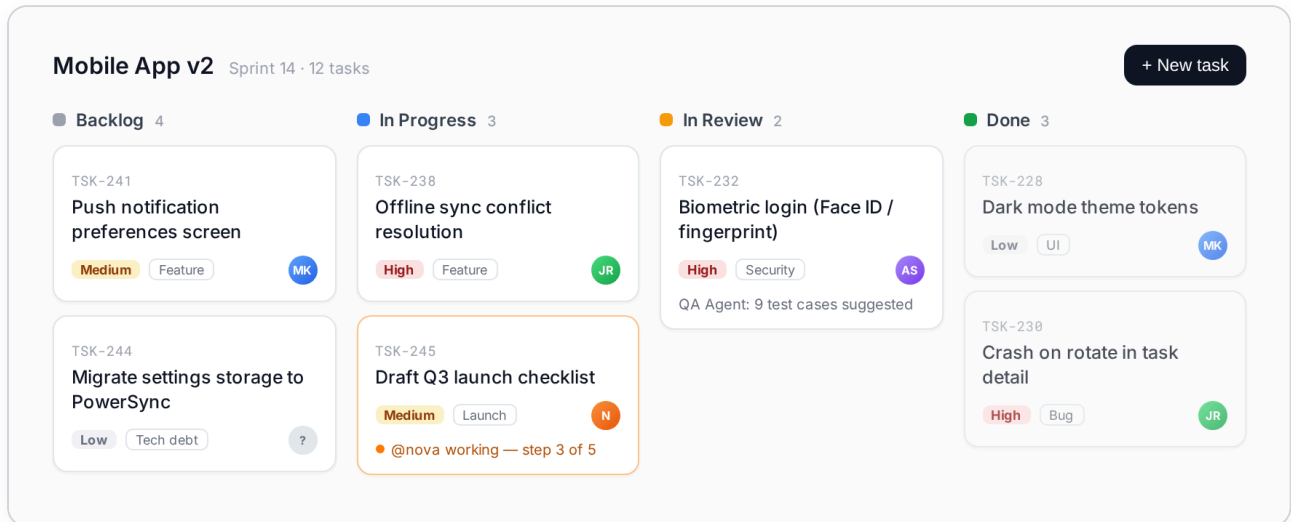
offline

local-first sync — fast on bad  
wifi

**In this paper:** the sprint board, the bug-tracking workflow, and the three agents — Sprint Planner, Bug Triager and QA Agent — that take the coordination work off the team. Screenshots are taken directly from the product.

## § Work, visible — including the agents'

Tasks carry types, states, priorities and assignees — and an assignee can be an agent. Agent work shows up on the same board as everyone else's, progress included.

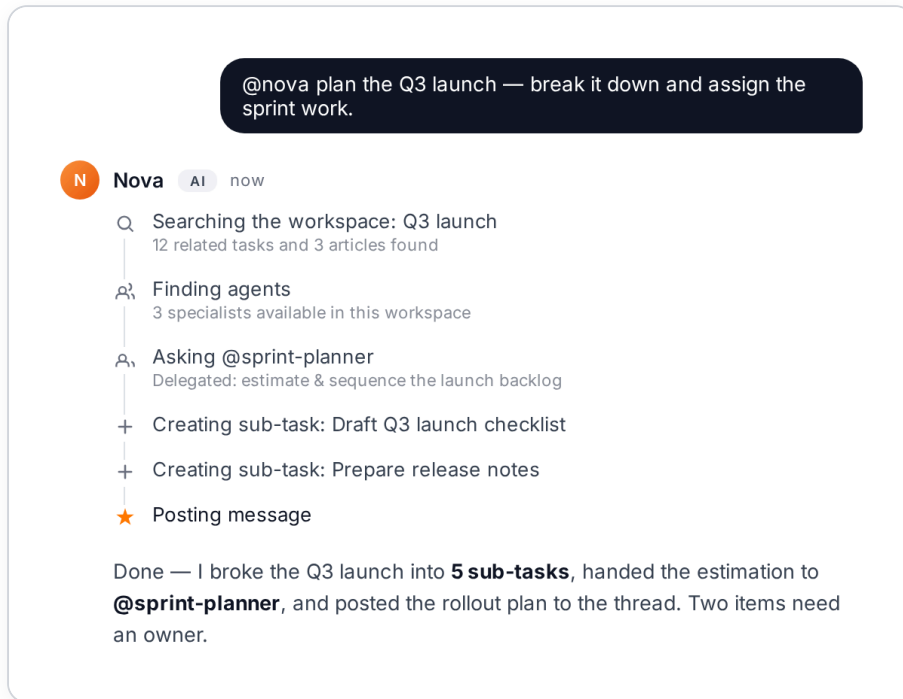


**A sprint in flight.** TSK-245 is assigned to @nova, which is three steps into drafting the launch checklist; TSK-232 already has nine QA-suggested test cases attached.

Workflows aren't one-size-fits-all: the workspace is seeded from templates — **Task Management, Bug Tracking, Product Development** — each adding task types, states and sample content, all editable afterwards. Real-time sync is local-first, so boards stay instant even when the connection isn't.

## § From vague request to sprint-ready

The slowest part of any sprint is the front of it: turning "we should launch in Q3" into work people can pick up. That's a delegation, not a meeting.



**One mention, a plan.** Nova searches the workspace for context, pulls in @sprint-planner for estimation, creates the sub-tasks and posts the rollout plan — every step visible as it runs.

The Sprint Planner template breaks large tasks into subtasks, suggests t-shirt sizing with justification, finds blocking dependencies via semantic search, and checks current assignments before proposing owners. It doesn't just advise — it **creates the subtasks**, so the backlog is ready when the standup starts.

## § Bugs triaged, edge cases caught

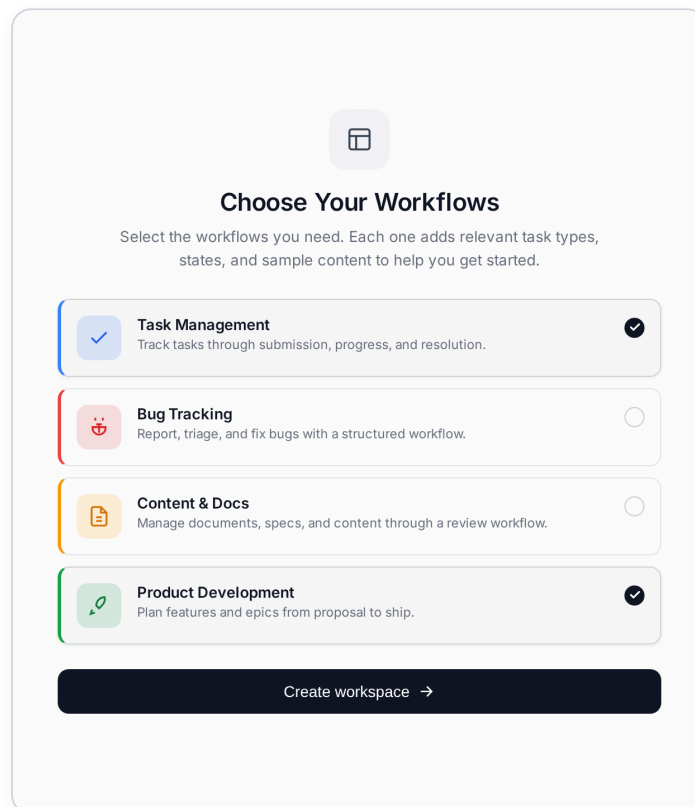
Two more specialists cover the unglamorous half of shipping: deciding what a bug report means, and asking the questions a hurried task description skipped.

### Bug Triager

Mention it on a bug and it assesses severity with justification, identifies the likely component, checks for duplicates via semantic search, sets the priority on the task, and flags what information is missing. Critical issues get escalated explicitly instead of sinking into the backlog.

### QA Agent

Pointed at a feature task, it reviews testability, generates concrete test cases — happy path, boundaries, error states, security — and flags ambiguous acceptance criteria. Specific by design: not "test the login", but "test login with valid credentials, invalid password, expired account, SQL injection in the email field."



**Workflows as templates.** Bug Tracking ships report → triage → fix states out of the box; Product Development covers proposal → ship.

## § Decisions that stay findable

The other half of the coordination tax is archaeology: why did we build it this way? Who decided? Where's the spec?

Because conversations, tasks and knowledge live in one workspace, the trail stays connected: a task links to the spec page, the spec links back to the tasks that implement it, and the thread where the decision happened is one click away. **Semantic search** spans all of it — agents use the same search, which is why their triage and planning is grounded in your team's actual history rather than generic advice.

- **Docs Writer** turns shipped tasks into knowledge-base articles, so documentation happens as a by-product of shipping.
- **Release Notes Writer** groups a release's tasks into themes and writes the customer-facing notes, linked back to the work.
- **Conversations** include channels, threads and voice/video calls — standups and incident calls happen next to the work they're about.

### A sprint week, compressed

MOMENT	WHO DOES IT
Epic lands as a one-liner	Nova + Sprint Planner break it down, size it, propose owners
Bug reports arrive overnight	Bug Triager sets severity, dedupes, escalates the critical one
Feature ready for review	QA Agent attaches test cases; humans review the diff
Release ships Friday	Release Notes Writer drafts the notes from the closed tasks

## § Autonomy with a paper trail

Agents that create and edit tasks need to be governable — and in Neuphlo, governance is platform-level, not a promise.

- **Every run is visible.** Agent work streams as a step-by-step timeline and is preserved with the result — no silent edits.
- **Confidence gates.** Owners set per-tool thresholds; low-confidence mutations require human confirmation before they land.
- **Scoped tools.** Each agent can only call the tools in its scope — the QA Agent reads and comments; it doesn't delete.
- **Your model, your data path.** BYOM means the team's code-adjacent context goes to the provider you chose — or stays entirely local on a self-hosted deployment with bundled Ollama.

## § Getting started

Create a workspace with the **Task Management** and **Bug Tracking** templates, connect an AI provider, and add Sprint Planner, Bug Triager and QA Agent from the template library — about ten minutes end to end. Then assign an agent its first task and watch the board.

**In short:** Neuphlo gives software teams one workspace where the routine coordination — breakdown, triage, test design, release notes — is done by governed agents, and the humans keep the judgement calls.