ASANATRACKER REVIEW

# Asana Sprint Tracking: Backlogs, Capacity, and Reports

Plan and track sprints in Asana with backlogs, sprint boards, capacity signals, dashboards, retrospectives, and tool limitations.

Saskia Linwood, Senior Editor · 28.02.2026

**TL;DR** Sprint tracking in Asana is a practical setup for teams running 1–3 week iterations without formal Scrum overhead. A Backlog project, a Sprint project (or sprint-bounded sections), custom fields for points and priority, and Workload on Advanced cover most agile-lite practice. The gaps versus Jira or Linear show up in native burndown charts, velocity reports, and Git-event automation. This how-to walks through sprint setup, capacity planning, execution tracking, sprint reports and retrospectives, and where dedicated software-team tools win. Feature availability verified May 20, 2026.

## Sprint Tracking Setup in Asana

**A clean setup uses two projects: a Backlog with ordered work and a Sprint project with the current iteration. Custom fields carry priority, story points, status, and acceptance criteria.**

Set up takes about an hour for the first sprint. Templates save time on subsequent sprints — saving the Sprint project as a template means each new iteration starts pre-populated.

1. **Create the Backlog project** — priority-ordered list view; add sections for Ready, Refined, Triaged
2. **Create the Sprint project** — board view with Backlog, In Progress, In Review, Done columns
3. **Add custom fields** — Story points (number), Priority (single-select), Type (single-select), Status (single-select)
4. **Pick the sprint duration** — 1–3 weeks; 2 weeks is the modal choice for cross-functional teams
5. **Save the Sprint project as a template** after the first iteration; clone it for each new sprint

Avoid one project per sprint if the team rotates members frequently. Date-bounded sections in a single Sprint project are easier to maintain.

*Two projects: Backlog + Sprint. Custom fields for points, priority, type, status. Save as template after sprint one.*

## Planning Sprint Capacity

**Capacity planning sums story points (or hours) per assignee against a per-sprint**

**capacity number. Workload view on Advanced does this visually; smaller teams can do it in a list with sum rollups.**

Capacity planning is where sprints usually fail. Pulling too much in is the default failure mode; the discipline of a hard capacity per person prevents it.

- **Workload inputs** — Story points custom field, per-person capacity (e.g. 8 points per 2-week sprint)
- **Estimates** — relative (Fibonacci) is more honest than absolute; absolute (hours) creates false precision
- **Assignee balance** — Workload view (Advanced) shows everyone\'s sum at a glance
- **Spillover** — incomplete tasks at sprint end auto-roll if the rule is set; otherwise the team decides per task
- **Buffer** — leave 20% of capacity unallocated for emergencies, interruptions, support work

Track planned-vs-completed points per sprint for 5+ sprints before drawing velocity conclusions. Two data points aren\'t a trend.

> *Capacity = points per person × sprint length × 0.8 buffer. Track velocity over 5+ sprints before trusting it.*

## Tracking Sprint Execution

**During the sprint, the board view is the daily working surface. Standups happen against the board; blocked work surfaces via a custom field; rules automate the routine moves.**

The board does the work. A clean board makes the standup take 5 minutes; a messy board makes it take 25 and surface less.

- **Daily standup** — walk the board right-to-left (Done → In Review → In Progress → Backlog); surface blockers first
- **Blocked items** — Blocked custom field; sort or filter to find them fast; assign an unblocker as part of the standup
- **Owner updates** — keep updates in task comments, not in chat; chat fragments context
- **Automation** — when section = In Review, assign to lead reviewer; when section = Done, notify QA
- **Recurring routines** — sprint kickoff task, mid-sprint check, retro task — all as recurring or templated

If the standup is a status meeting instead of an unblocking meeting, the format isn\'t working. Walk the board, ask about blockers, end early.

> *Walk the board right-to-left, surface blockers first, end early. Automate the routine moves.*

## Sprint Reports and Retrospectives

**Sprint reports and retrospectives close the loop. Dashboards show committed-vs-completed, point burnup, and blocker patterns; retros consume that data to improve the next sprint.**

The two reports that matter: closed-vs-committed (did we finish what we said we would?) and blocker patterns (where did we lose time?). Everything else is supporting context.

- **Sprint progress dashboard** — burnup of completed points, count of in-progress vs completed, blocker count
- **Closed sprint review** — total points completed, spillover count, blockers resolved, blockers persisting
- **Retrospective questions** — what worked, what didn\'t, what to try next; 30–45 minutes max
- **Action items** — retro outcomes become tasks in the next sprint; track completion to make retros stick
- **Velocity tracking** — store closed-points-per-sprint in a separate Goals page or spreadsheet for trend

Retros without follow-through become rituals. Make retro action items first-class tasks in the next sprint.

> *Closed-vs-committed + blocker patterns. Make retro action items tasks in the next sprint.*

## Limitations Versus Jira or Linear

**For software-only teams running formal Scrum, Jira and Linear close gaps Asana can't. Burndown, velocity, Git-event automation, and JQL queries are the most-missed features.**

Recognise the limit before working around it. Building burndown charts manually in Asana wastes more time than switching tools, if velocity is a critical metric.

- **Missing native burndown** — workaround exists; takes manual upkeep
- **Story points as second-class data** — workable, but Jira treats them natively
- **Git event automation** — Linear is best-in-class; Jira is solid; Asana has integrations but they\'re lighter
- **Advanced sprint reports** — cumulative flow, cycle time, lead time — minimal in Asana; deep in Jira
- **Integrations that close the gap** — Linear or Jira sync, Everhour for time, third-party burndown apps

For mixed teams that share a workspace with marketing and design, Asana\'s sprint tracking is usually good enough. For dev-only teams, switching costs are worth paying.

> *Burndown, velocity, Git events — Jira/Linear win. For mixed teams, Asana's sprint tracking is sufficient.*

## FAQ

**How long should a sprint be in Asana?**
One to three weeks. Two-week sprints are the most common choice for cross-functional teams. Shorter sprints suit teams iterating on tight feedback loops; longer sprints suit teams with heavy review or QA cycles. Pick one cadence and stick with it for at least five sprints before changing.

**Should I create one Sprint project per sprint or use sections?**
For teams with rotating members or frequent reorganisation, use date-bounded sections in a single Sprint project — easier to maintain. For stable teams, one project per sprint gives cleaner archives. Either pattern works; consistency matters more than the choice.

**How do I track velocity in Asana?**
Manually. Add a number custom field "Story points", sum completed points per sprint, and log the total in a separate Goals page or spreadsheet. Track over 5+ sprints before drawing conclusions; fewer data points create false trends. Jira and Linear track velocity automatically.

**What happens to incomplete tasks at sprint end?**
A rule can auto-roll them to the next sprint, or you can decide per task during sprint planning. Most teams default to rolling them and then asking "should we still do this?" — sometimes the answer is no, and the task gets dropped instead of carried over.

**Can I run a daily standup against an Asana board?**
Yes. Open the Sprint project in board view and walk right-to-left (Done → In Review → In Progress). Surface blockers first; the goal is unblocking, not status reporting. Keep standups under 15 minutes by ending when blockers are addressed rather than going through every team member.

---

Full article: https://asanatracker.com/asana-sprint-tracker