

Asana Agile Tracking for Scrum and Kanban Teams

Evaluate Asana for agile tracking, including backlogs, sprints, Kanban boards, reports, integrations, and Jira-style limits.

Saskia Linwood, Senior Editor · 16.04.2026

TL;DR Asana works for agile teams that practice Scrum or Kanban lightly — backlog, current sprint, in-progress, done — but lacks the depth of Jira, Linear, or GitHub Projects for formal Scrum, story points as first-class data, native burndowns, or Git event integration. Where it shines is teams where engineering shares a workspace with marketing, design, and PM: the same custom fields and dashboards work for sprints, campaigns, and launches in parallel. This page covers when Asana fits agile teams, how to model backlog and sprint planning, Kanban board workflow, reporting and dashboards, developer integrations, and the points where Jira or Linear is a better answer. Feature availability verified May 20, 2026.

Can Asana Work for Agile Teams?

Asana fits agile teams that prioritise cross-functional visibility over Scrum-specific depth. Kanban and lightweight Scrum work; formal Scrum at scale doesn't.

The deciding question is "how much agile theatre do you need?" Teams that run formal ceremonies, story-point estimation, velocity tracking, and burndown charts as a daily practice will fight Asana. Teams that work in sprints but stay flexible about ceremonies will be happy.

- **Scrum-lite** — backlog, sprint, in-progress, done; ceremonies optional — fits Asana well
- **Kanban** — board view with WIP limits as custom field convention — fits Asana well
- **Hybrid (Scrumban)** — sprint cadence with continuous-flow board — fits Asana well
- **Formal Scrum or SAFe** — story points, velocity, burndown, retros as artifacts — Jira fits better
- **Best fit by team maturity** — teams under 100 people that mix engineering with non-engineering work

Don't pick Asana for agile if half the team will leave anyway. Pick Linear if speed is the priority; pick Jira if formality and ceremony are.

Asana fits Kanban and Scrum-lite. Choose Jira or Linear for formal agile or velocity-focused workflows.

Backlog and Sprint Planning

A backlog in Asana is usually a separate project (or a section within the main project) that holds prioritised work. Sprints are date-bounded sections or separate

projects that pull from the backlog.

The two-project pattern (Backlog + Current Sprint) is the cleanest setup. Tasks move from one to the other via multi-homing during sprint planning.

- **Backlog project** — ordered list of work; priority, effort, and acceptance criteria as custom fields
- **Story points** — number custom field; not first-class like Jira, but workable
- **Sprint project** — date-bounded; either a new project per sprint or sections inside one Sprint project
- **Intake** — Form attached to the backlog; required fields enforce minimum quality on every submission
- **Capacity planning** — Workload view on Advanced; alternatively, sum story points per assignee in a list

If the team rotates sprint duration, a single Sprint project with date-bounded sections is easier to manage than one project per sprint.

Backlog + Sprint projects, story points as number custom field. Workload (Advanced) for capacity.

Kanban Boards and Workflow Stages

Asana's board view supports the classic backlog → ready → in progress → in review → done flow. WIP limits aren't a first-class feature but can be enforced via custom field convention plus a rule.

For pure Kanban teams, Asana's board view is competent. The missing piece is native WIP limit enforcement; the workaround is a rule that flags overlimit columns.

- **Columns** — typically Backlog, Ready, In Progress, Code Review, QA, Done
- **WIP limits** — track via a count and a rule; not built into the UI but enforceable
- **Blocked work signals** — Blocked custom field; show as a coloured field on the card
- **Card design** — show owner, priority, and one effort field per card; more becomes noise
- **Automation** — when section moves to Code Review, assign to lead reviewer; auto-archive 14 days after Done

Kanban discipline matters more than the tool. A clean board in Asana beats a sloppy board in Jira; tooling doesn't fix process problems.

Board view fits Kanban. Workaround WIP limits with a rule + custom field count.

Agile Reporting and Dashboards

Agile reporting in Asana relies on dashboards and Universal Reporting (Advanced) rather than native sprint reports. Velocity and burndown require manual setup or a third-party connector.

The reporting gap is the most-cited Asana-vs-Jira complaint. Several workarounds exist; none matches Jira's built-in sprint reports.

- **Progress views** — completion-over-time chart in the sprint project; closes-vs-opens ratio
- **Burndown proxies** — bar chart of remaining story points by day; manual setup, refresh via dashboards
- **Sprint health** — closed-vs-committed ratio; track over 5+ sprints for a velocity feel
- **Velocity** — sum of points completed per sprint; track in a separate Goals or spreadsheet view
- **Retro inputs** — list of completed tasks per sprint with notes field for retro discussion

If the team genuinely needs burndown and velocity reports as a primary metric, Jira will save them weeks of dashboard tinkering.

Universal Reporting + manual burndown proxies. If velocity is critical, Jira wins.

Integrations for Developers

GitHub, GitLab, Bitbucket, Jira, Slack, and Microsoft Teams cover most developer integration needs. Two-way Jira sync is available but not perfect.

Developer integrations matter for two reasons: linking commits to tasks for audit trails, and avoiding double entry into both Asana and a code platform.

- **GitHub / GitLab** — link branches and PRs to tasks; commit messages can update task status via integration
- **Jira sync** — two-way for some teams that run mixed-tool environments; verify field mapping before relying on it
- **Slack / Teams** — notifications for assignments, completions, and status changes
- **API and webhooks** — for systems not in the official catalogue; rate limits documented in developer docs
- **When a dev-native tool wins** — code-only teams; deep Git workflow; advanced branching strategies

If developers spend more time in their IDE than in Asana, push tasks into a dev-native tool and use Asana for coordination instead.

GitHub/GitLab plus Slack covers most dev needs. Jira sync exists but is fiddly — verify field mapping.

Agile Tracking Limits and Alternatives

For pure software teams, Jira or Linear usually wins. For mixed teams that want sprint cadence without ceremony, Asana is a defensible choice.

The decision rarely hinges on features alone. It hinges on whether the team needs deep agile artefacts or whether speed and cross-functional visibility matter more.

Need	Right tool
Cross-functional team running sprints	Asana Advanced
Modern software team prioritising speed	Linear
Formal Scrum or SAFe at scale	Jira
Open-source-style flat workflow	GitHub Projects
Tiny team experimenting with Kanban	Trello or Notion board

- Asana's biggest agile weakness is reporting depth; if that's the priority, Jira saves time long-term
- Asana's biggest agile strength is keeping engineering and marketing in the same workspace
- Migration cost from Asana to Jira (or vice versa) is real — expect 4–8 weeks of friction

Run a parallel pilot for two weeks before committing. Feature checklists lose to lived experience every time.

Asana for mixed teams running sprint cadence. Jira for formal Scrum, Linear for speed-first software teams.

FAQ

Does Asana support story points?

Not as a first-class feature, but easily as a numeric custom field. Most agile teams using Asana create a "Story points" custom field and reference it in Workload (Advanced) and dashboards. Jira treats story points as built-in data; Asana treats them as user-defined.

Can I run Scrum in Asana?

Yes for Scrum-lite practice — backlog, sprint, ceremonies as recurring tasks. Less well for formal Scrum at scale where velocity, burndown, sprint reports, and story-point estimation drive daily decisions. For that, Jira and Linear go deeper.

Does Asana have a burndown chart?

Not natively. You can build a burndown proxy with a custom dashboard card that tracks remaining story points by day across a sprint, but it requires manual setup and refresh. Jira ships native burndown and velocity reports; that's a meaningful gap.

How does Asana compare to Jira for agile teams?

Asana wins on UX polish, cross-functional visibility, and onboarding speed. Jira wins on agile artefact depth: native burndown, velocity, advanced roadmap, JQL queries, and tighter Git integration. The choice usually comes down to whether the team is software-only or mixed.

Can I sync Asana with Jira?

Yes, via the Jira Cloud integration. The sync is two-way for some teams running mixed-tool environments. Verify field mapping carefully — custom fields, story points, and complex workflows can desync. Most teams pick one tool as the source of truth and use the other read-only.

Is Asana a Jira alternative?

For mixed and non-engineering-heavy teams running sprint cadence, yes. For software-only teams or organisations practising formal Scrum/SAFe at scale, Asana lacks the agile depth Jira provides. The realistic alternative comparison list also includes Linear, ClickUp, and Azure DevOps.

Full article: <https://asanatracker.com/asana-agile-tracker>

AsanaTracker may earn a commission when readers sign up to Asana through links on this site. Editorial decisions stay independent.