Phillip Anderson 2600240479-5
Yew Zhi Jie 2600240495-7

Jira Research Report

Introduction

Jira is Atlassian's work-management and issue-tracking platform with first-class Agile boards, reports, automation, and a large app marketplace. It's available primarily as a cloud service (with REST APIs and native dev-tool integrations) and is broadly used across enterprises.

Redmine is a free, open-source project/issue tracker built on Ruby on Rails. Out of the box it supports multi-project management, flexible workflows and permissions, Gantt & calendar views, time tracking, per-project wikis/forums, multiple databases, and 49 UI languages. It is typically self-hosted (though third-party hosting exists). Licensed under GPLv2.

Similarities

- **Core issue tracking & workflows** (custom statuses/fields, role-based permissions).
- **Multi-project management.**
- **APIs for integration/automation:** Jira (Cloud & Data Center) has extensive REST APIs; Redmine exposes JSON/XML REST endpoints.
- **Notifications & feeds** (email/RSS) and **time tracking** capabilities.
- **Plugin ecosystems** to extend the base product (Atlassian Marketplace / Redmine Plugins directory).

Differences

A. Licensing & Cost

- **Jira (Cloud)**: Proprietary; Free plan (up to 10 users), then paid tiers (e.g., Standard, Premium, Enterprise) on a per-user basis. Pricing is tiered and varies by region.

- **Redmine**: Open source, **no license fees** (GPLv2). You still bear hosting/ops costs; many teams use commercial hosts if they don't want to run it themselves.

B. Deployment model

- **Jira**: Cloud is the focus. Atlassian announced it is **sunsetting Data Center** products on a phased timeline:
  • **Mar 30, 2026** – end of new Data Center subscriptions for new customers
  • **Mar 30, 2028** – last date for existing customers to purchase new DC licenses/apps/expansions
  • **Mar 28, 2029** – Data Center end-of-life (licenses become read-only), with security updates provided through the period.

  Cloud scale has grown and **Jira Cloud now supports up to 100,000 users** on a single site.

- **Redmine**: Designed for **self-hosting** (Linux/Unix/macOS/Windows; MySQL/PostgreSQL/SQLite). Turn-key installers and managed Redmine hosting are available from third parties (e.g., Bitnami images; Planio).

C. Agile capabilities

- **Jira**: Scrum & Kanban boards are **built-in**, with sprint/board reports (Burndown, Velocity, etc.).

- **Redmine**: Core includes Gantt & calendar; **Agile boards/charts typically require plugins** (e.g., RedmineUP Agile or community Kanban).

D. Automation & query language

- **Jira**: Visual **Automation** rules (Cloud) and **JQL** advanced search are core differentiators.

- **Redmine**: No native equivalent to JQL/Automation; extensibility achieved via plugins, custom fields, and API scripts.

E. Integrations & ecosystem

- **Jira**: Deep, supported integrations with **GitHub/GitLab**, Bitbucket, CI/CD, and thousands of Marketplace apps (Atlassian cites **5,700+** apps).

- **Redmine**: Built-in SCM viewing for Git/Subversion/etc.; many third-party plugins add Agile, CRM, helpdesk, etc. (quality varies; often needs admin effort).

F. Security/compliance & identity

- **Jira Cloud**: Enterprise SSO (SAML) and **SCIM** user provisioning via **Atlassian Guard**; data residency options are expanding.

- **Redmine**: As self-hosted software, security & compliance are **your team's responsibility** (server hardening, backups, SSO via plugins/proxies if needed). (General capability per Redmine architecture and plugin model.)

When to Choose One Over Another

Choose Jira when you need:

- **Managed cloud at scale** with native Agile boards/reports, and minimal ops overhead. (Built‑in Scrum/Kanban, Burndown/Velocity, 100k‑user scale.)
- **Turn‑key integrations** (GitHub/GitLab/CI/CD), plus a very large **Marketplace** of supported apps.

- **Enterprise controls** (SSO/SAML, SCIM provisioning, auditability) and a published cloud roadmap/compliance posture.

- A vendor‑led path forward (notably as Atlassian phases out Data Center in favor of Cloud).

Choose Redmine when you need:

- **Full control / on‑premise** deployment with **no license fees** and the ability to modify the source code.

- A **lightweight core** (issues, time tracking, wiki, Gantt) you can extend with plugins, and you're comfortable owning server maintenance (or contracting a Redmine host).

- **Multi‑language** teams (49 languages) or specific DB/OS requirements.

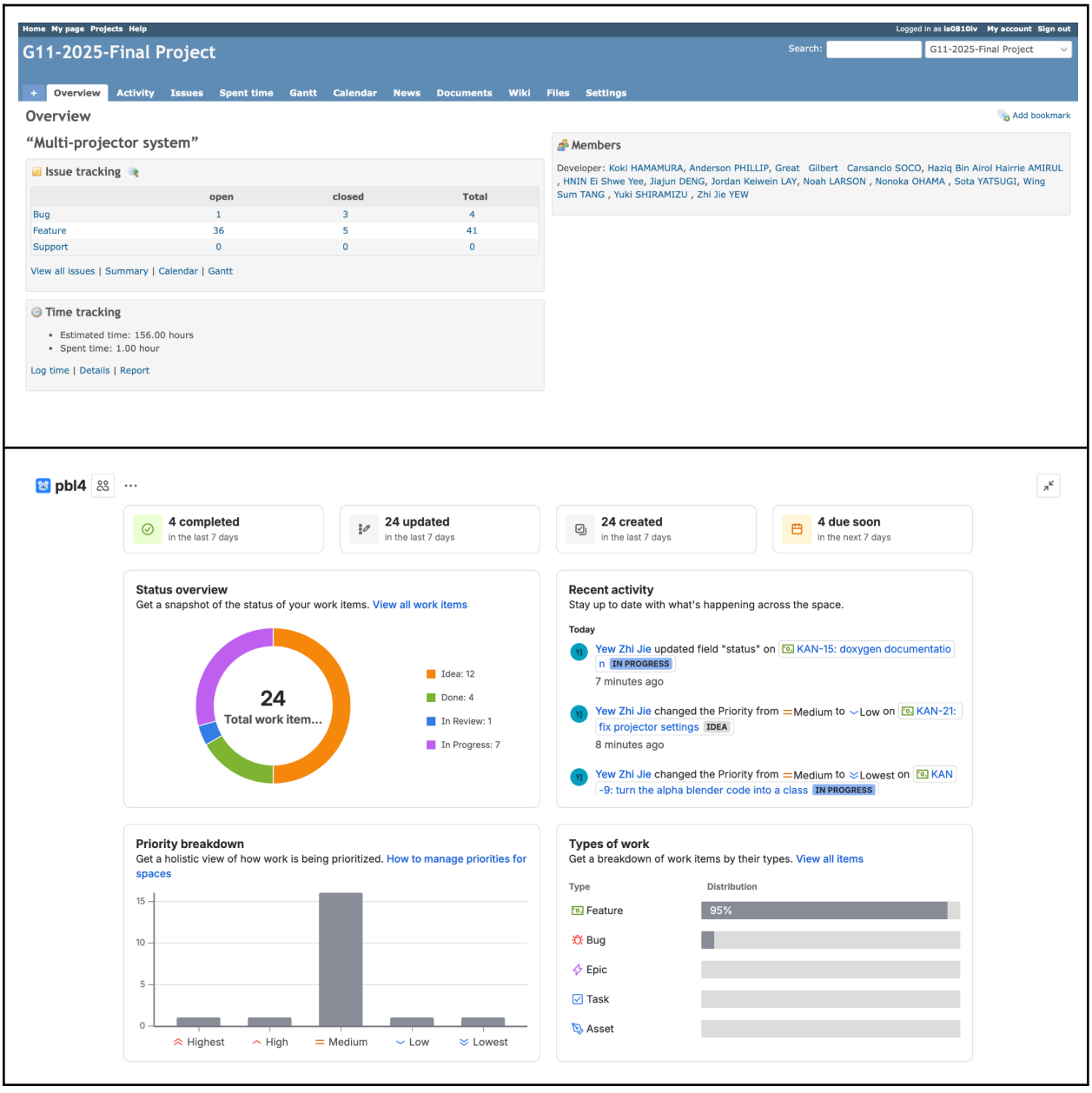
Total Cost of Ownership (TCO):

- **Jira**: Predictable subscription costs + paid Marketplace apps as needed; low infra/ops overhead in Cloud.

- **Redmine**: $0 license, but expect infra/DevOps time, upgrades, backups, and possibly commercial plugins or managed hosting.

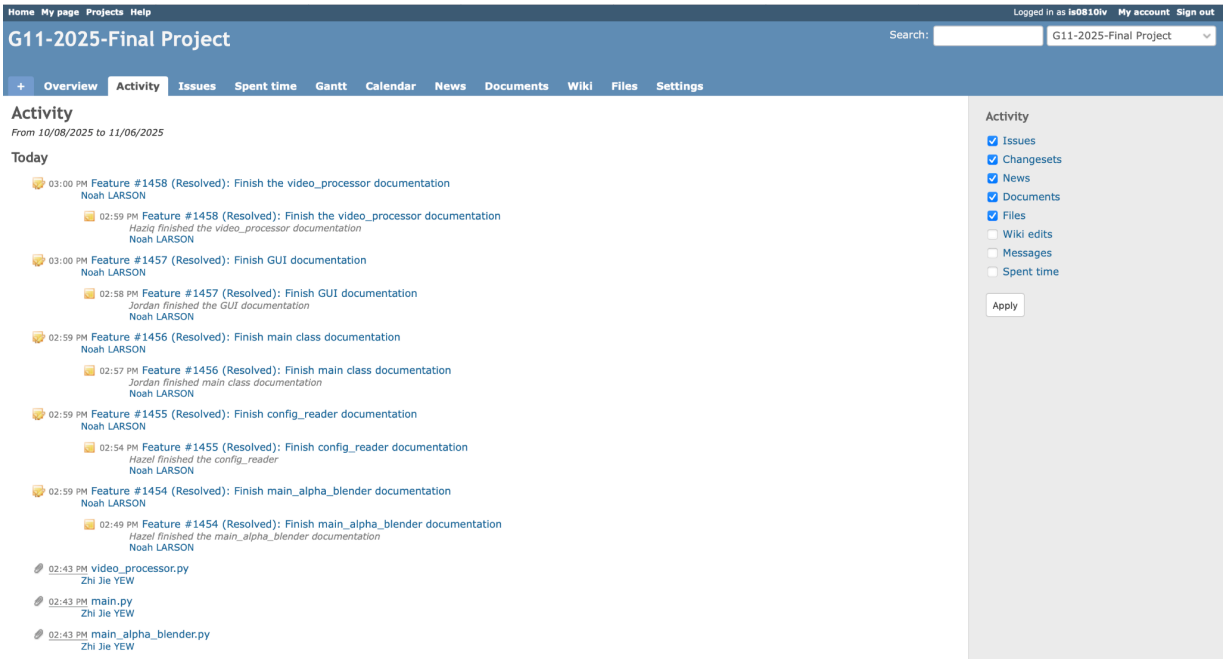
Jira vs. Redmine: Side-by-Side Comparison

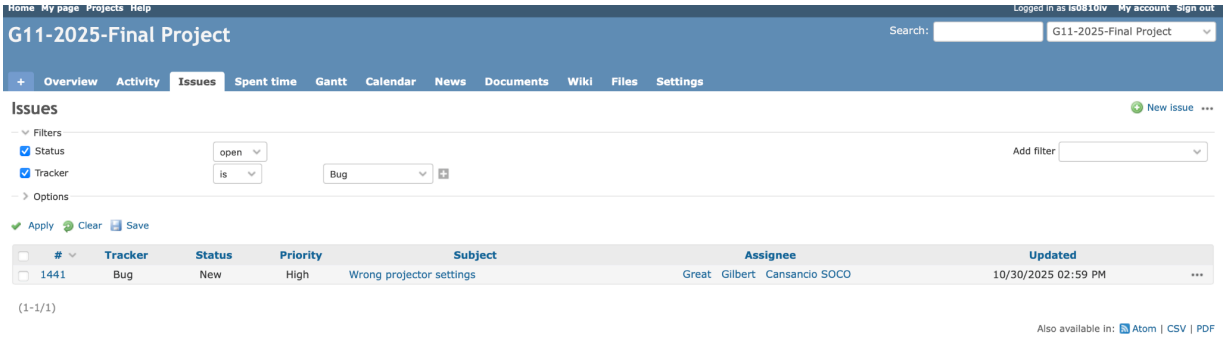
1. Redmine’s Overview page vs Jira’s Summary page

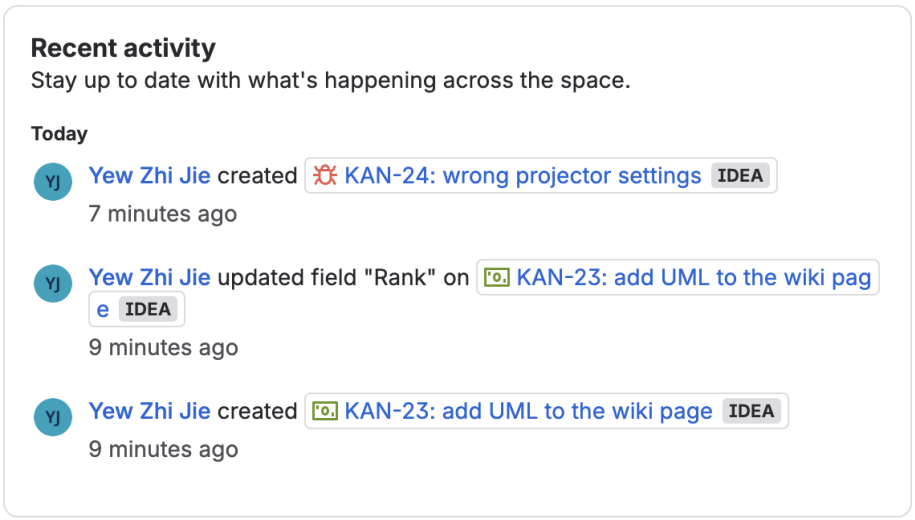


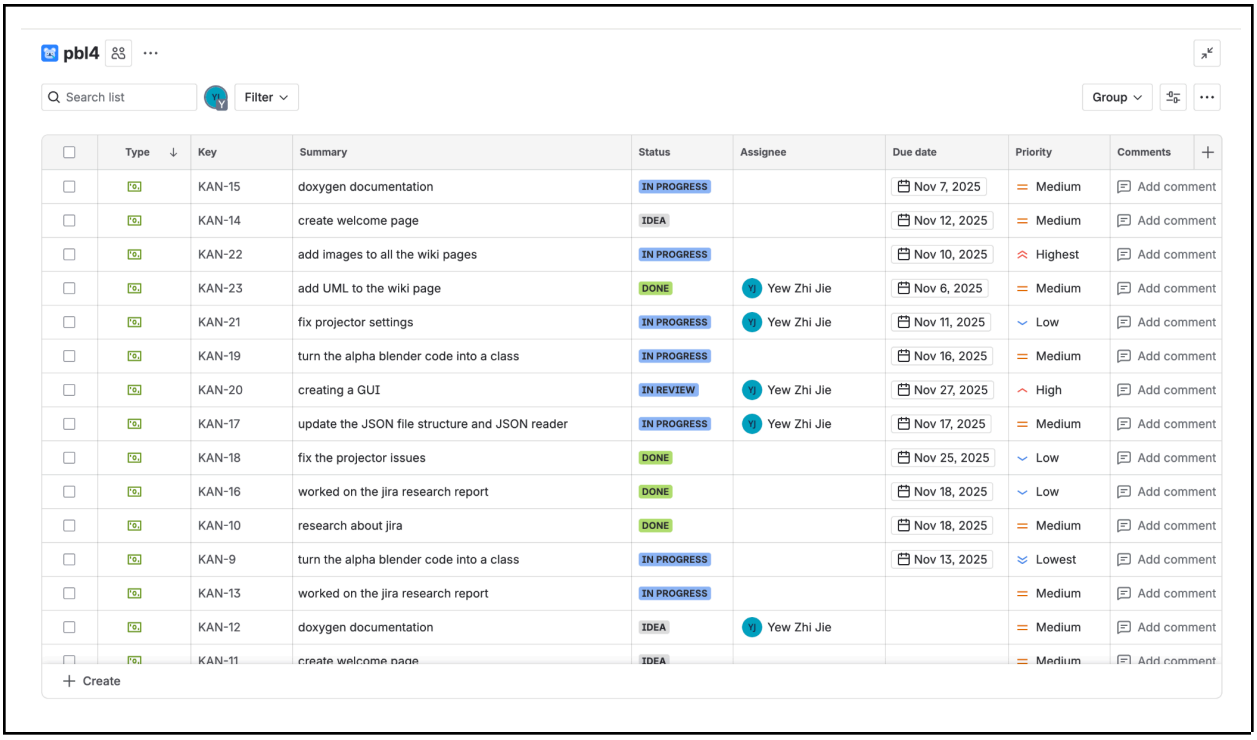
Redmine’s overview page is static and text based, showing only issue tracker, members, and time tracking. Jira’s summary is more visual with customizable widgets showing tasks progress, recent activities, priority breakdown, etc.

2. Redmine's Activity & Issues page vs. Jira's List page





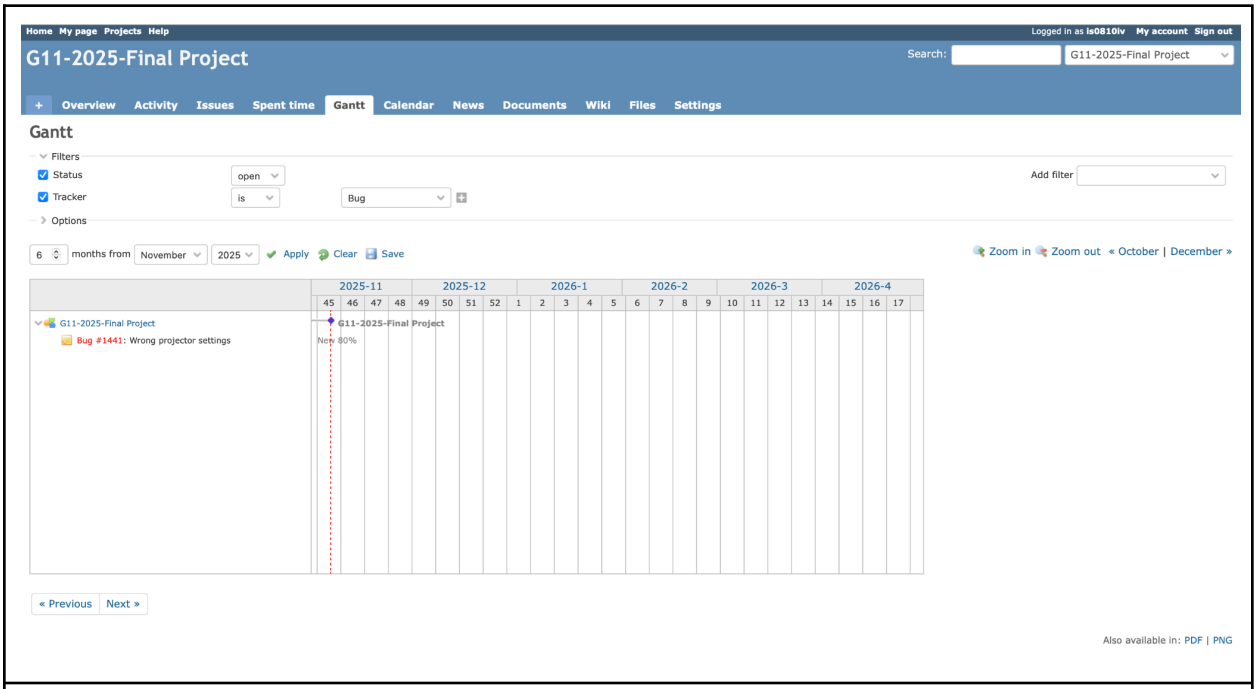


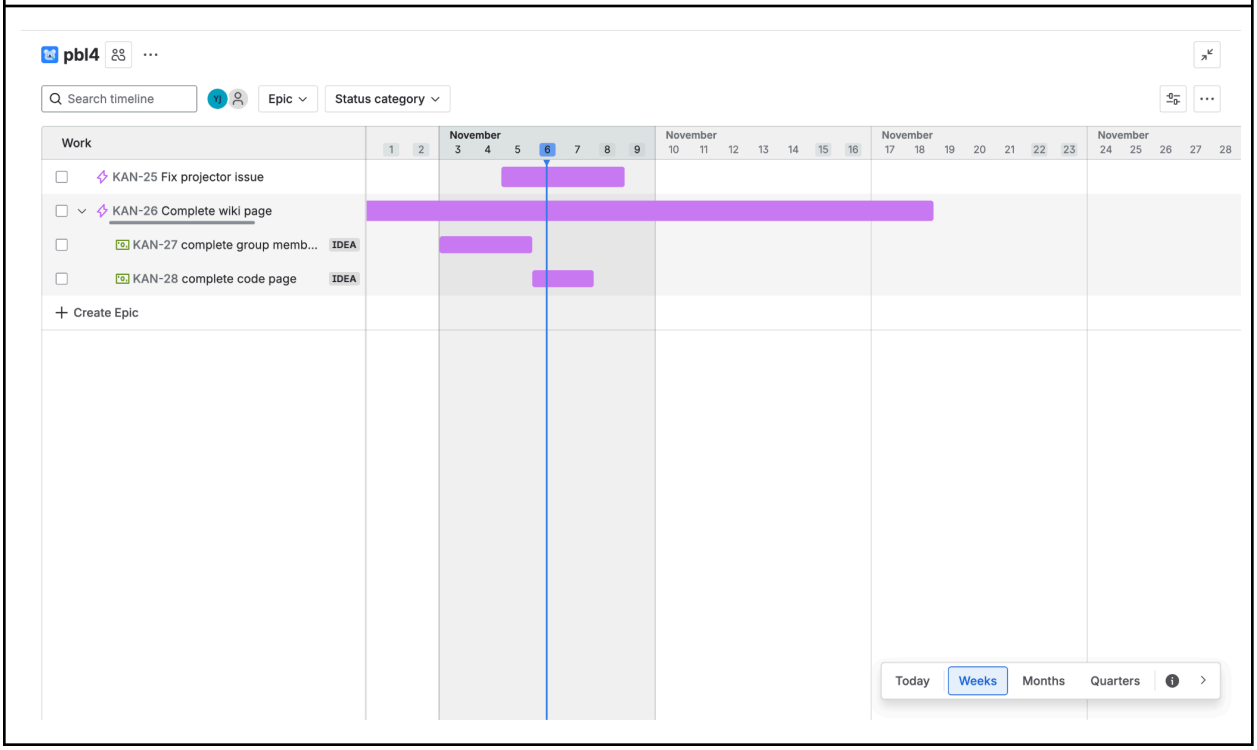


Redmine has Activity and Issues page separated, with Activity page showing timeline of all recent activities such as issue updates, and using Issues page to track, create, view, search, filter issues.

In Jira, there is a Recent Activities widget on the Summary page that shows all recent activities, similar to the Activity page on Redmine. There is also a List page which combines both activities and issues, showing all tasks, issues, bugs, etc. and also lets you track, create, view, search, and filter them. You can also change priority, due date, and add comments. Jira has a more user-friendly and intuitive user interface.

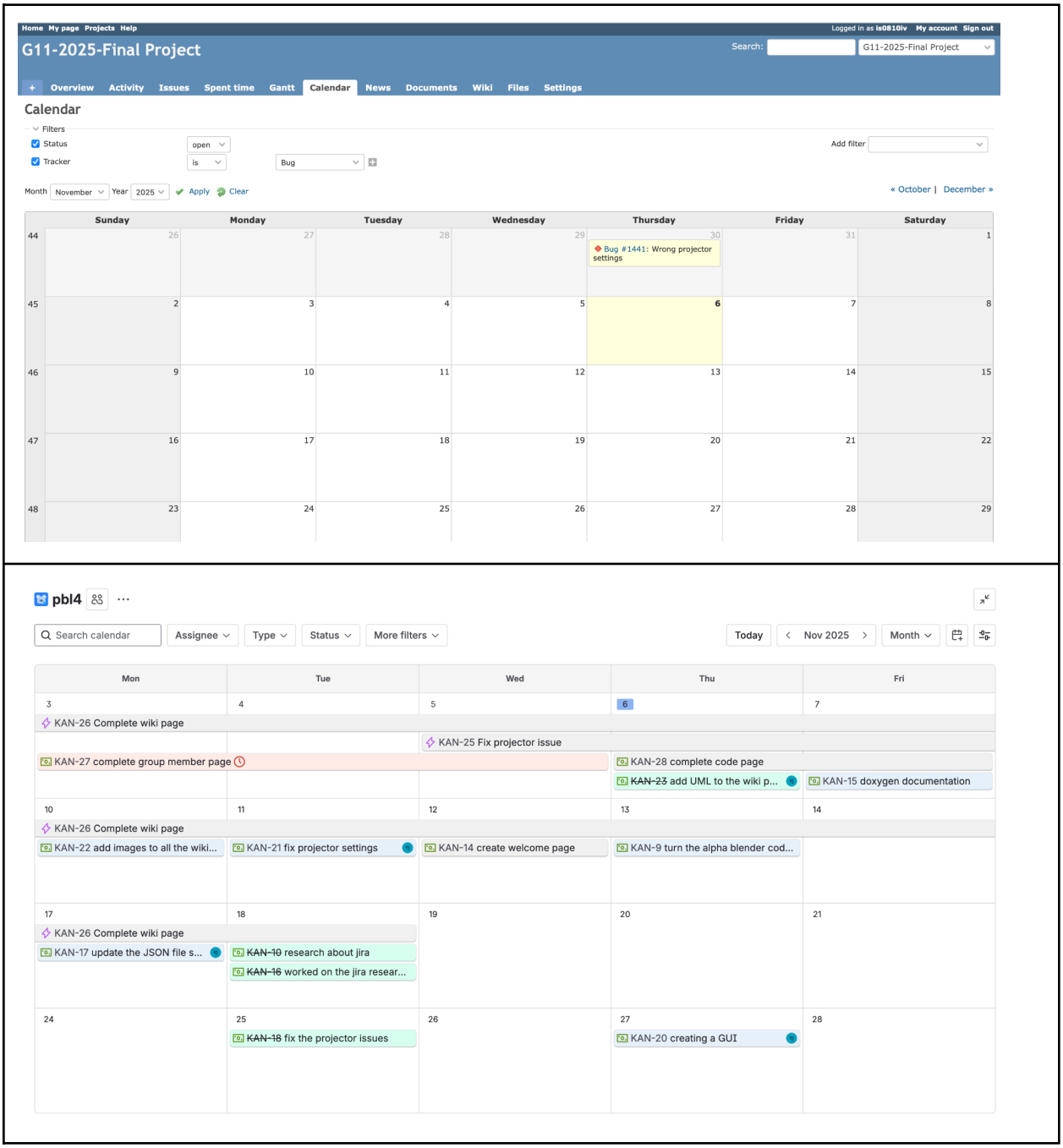
3. Redmine’s Gantt page vs. Jira’s Timeline page





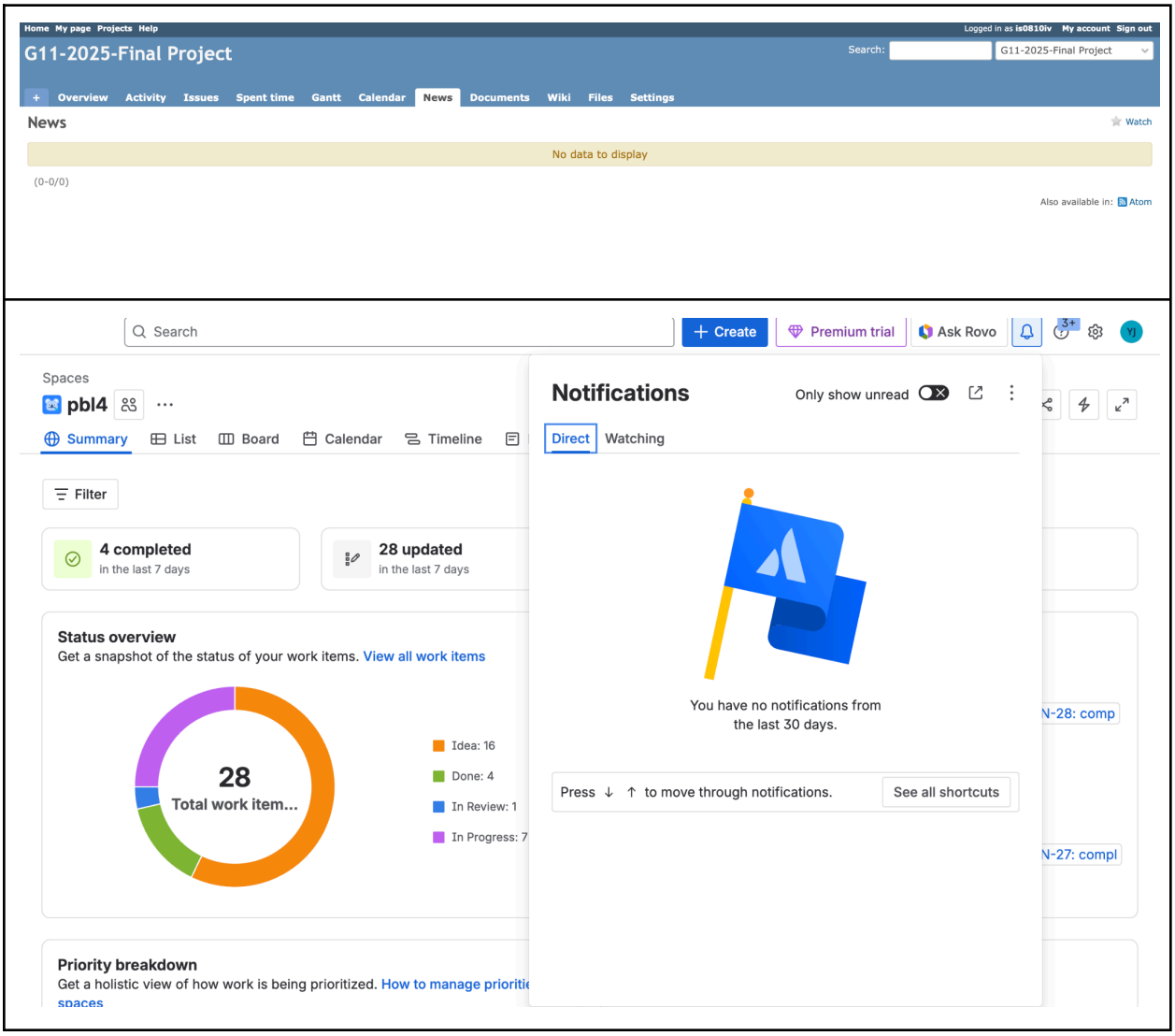
Redmine has a Gantt Chart page which displays issues on a timeline by start/due dates. Jira’s Timeline page is similar, but also shows Features, Tasks, Bugs, and you can also create tasks directly on the Timeline page.

4. Redmine’s Calendar page vs. Jira’s Calendar page



Redmine’s Calendar page shows a monthly layout of bugs and features, but only one at a time. Jira’s Calendar page is similar but shows all work types, including bugs, features, assets, epics, subtasks, and you can choose to filter for certain types.

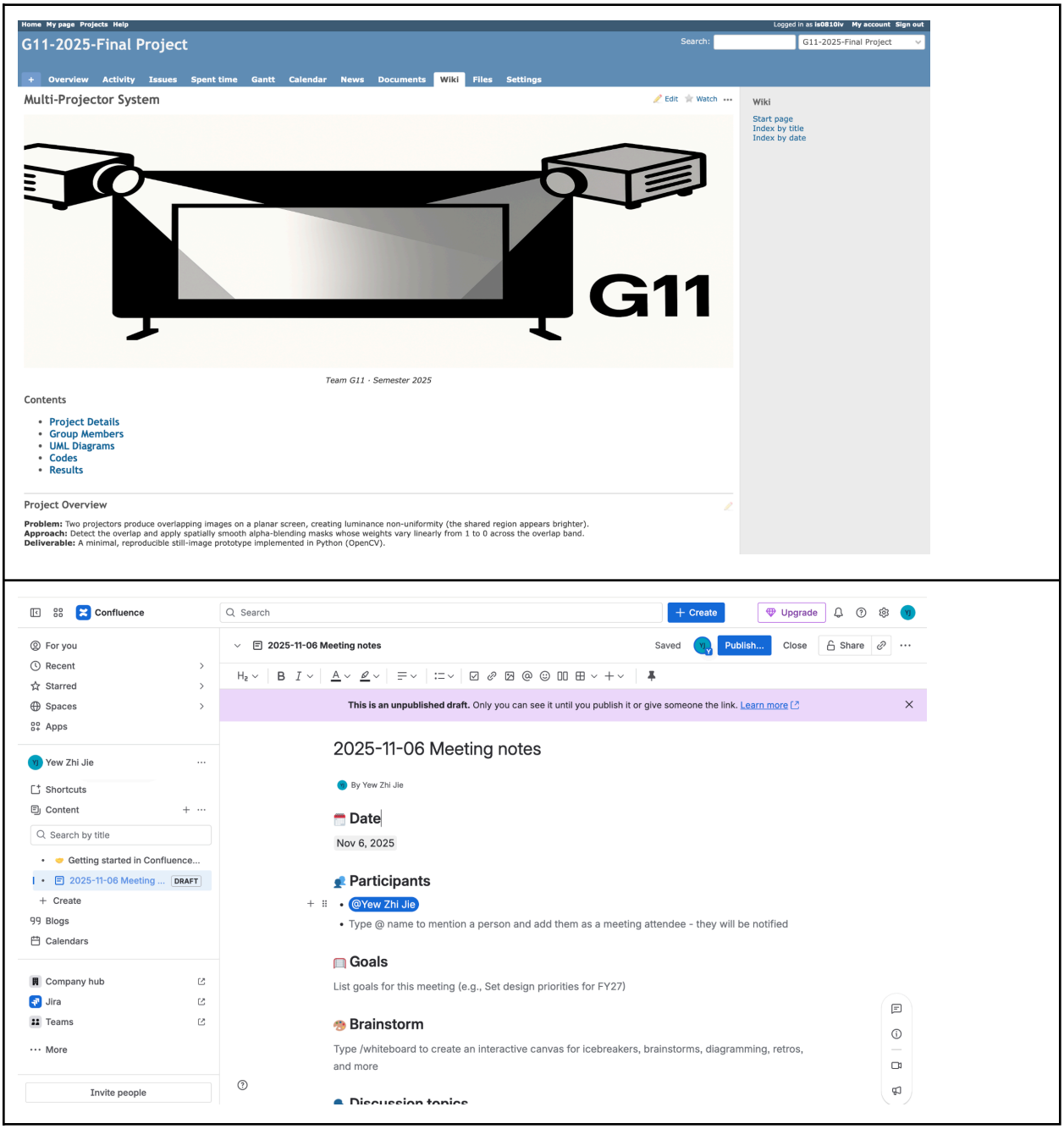
5. Redmine’s News page vs. Jira’s Notification tab



Redmine has a News page for posting updates, announcements, and change logs. However, you have to manually post.

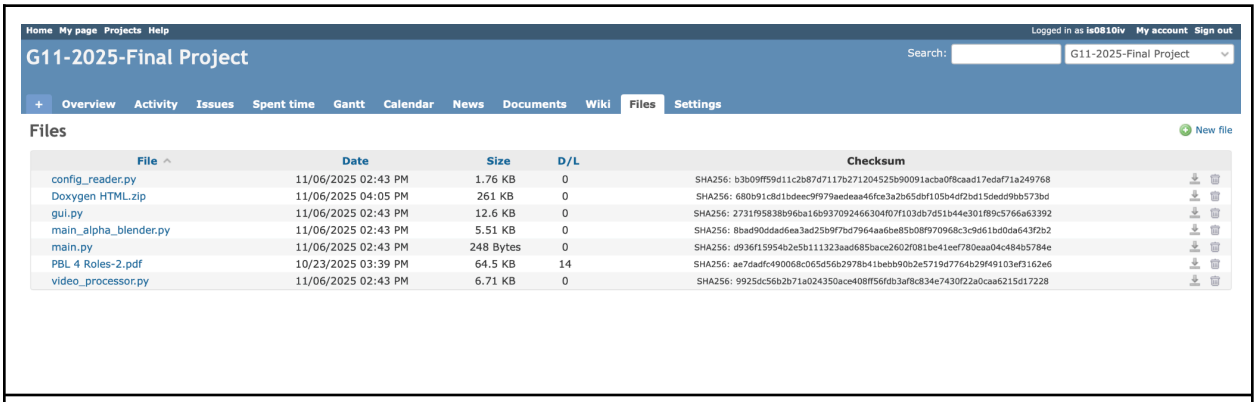
Jira integrated this feature as a Notification tab in the Summary page. It also automatically sends notifications when a deadline is near or when a work item is overdue.

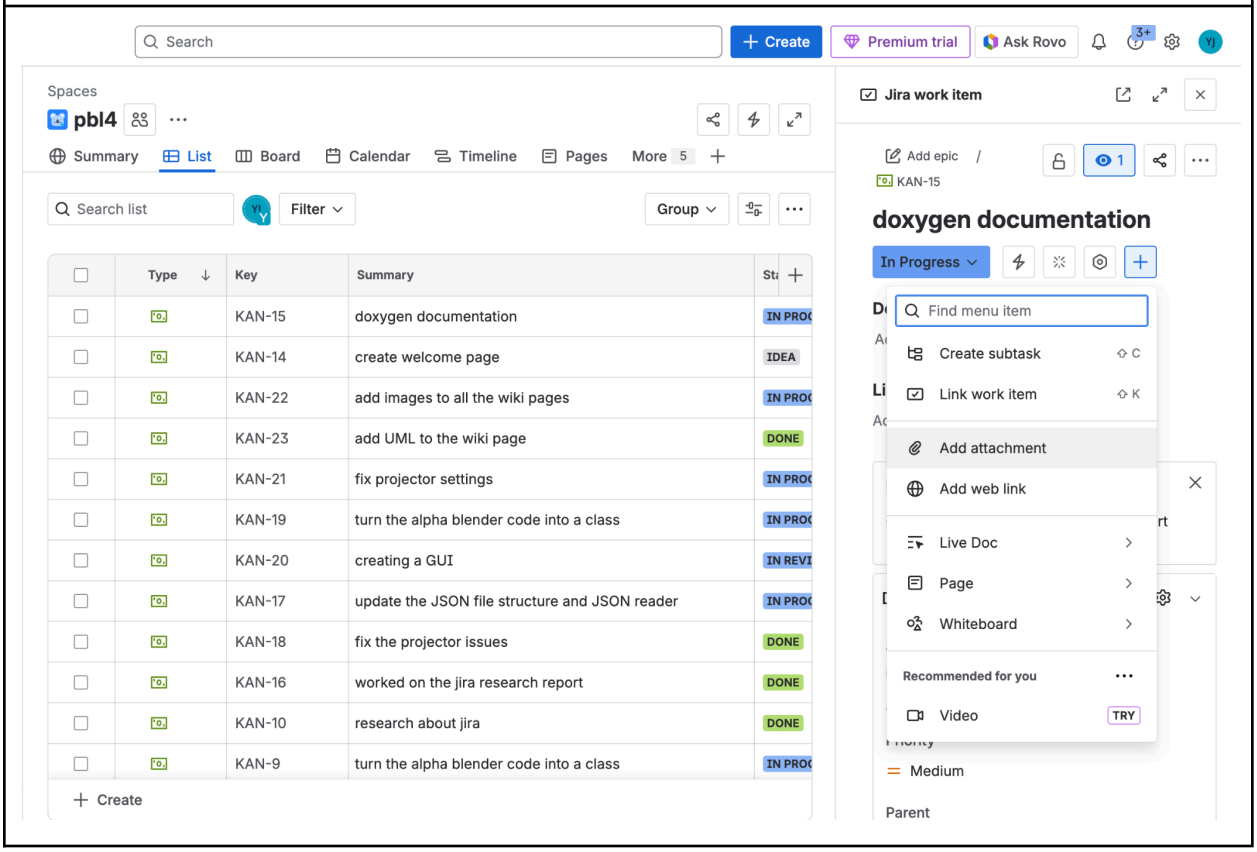
6. Redmine’s Wiki page vs. Jira’s Confluence page



Redmine’s wiki page is a lightweight, text based documentation tool that requires users to format content using markup languages like HTML. Jira’s equivalent for this is called Confluence, which offers a modern, “what you see is what you get” editor similar to Google Docs, allowing users to type, format, and embed media directly without needing markup.

7. Redmine’s File page vs. Jira’s issue-based file attachment





Redmine has a dedicated File page which lets you upload and manage all files in one centralized location.

Jira does not provide a standalone file repository. Instead, files are uploaded and stored as attachments within individual work items, meaning each file is tied to a specific task, bug, or feature.

Comparison Table

	REDMINE	JIRA
Primary Focus	General-purpose project and issue tracking	Agile project management (Scrum, Kanban, etc.)
Use Case	Broader use: software, IT, research, operations	Software development, DevOps, agile teams
Customization	Extremely flexible; open-source, so you can modify almost any part	High-level customization but within Jira's framework (limited by Atlassian's structure)
Workflow Editing	Basic workflow customization via configuration files or admin UI	Visual workflow editor (drag-and-drop transitions, conditions, validators)
Plugins/Add-ons	Community-driven plugins (free, open-source)	Huge marketplace (Atlassian Marketplace) with many paid add-ons
License	Open-source (free, GNU GPL v2)	Proprietary (paid, subscription-based)
Deployment	Self-hosted by default; some third-party cloud providers	Cloud (Atlassian Cloud) or self-hosted (Jira Data Center)
Cost	Free (you only pay for your hosting/server)	Requires payment per user/month
Issue Tracking	Standard issue tracking (custom fields, statuses, priorities)	Very detailed (custom issue types, subtasks, transitions)
Agile Boards	Needs plugins like Redmine Agile	Built-in Scrum and Kanban boards
Time Tracking	Built-in but simpler; can be extended	Built-in with reports
Gantt Charts / Roadmaps	Via plugins (e.g., redmine_gantt)	Built-in
Wiki / Forums	Built-in wiki and forum for collaboration	No native wiki, but integrates with Confluence
Integration	Integrations available but more manual setup	Strong integration with GitHub, Bitbucket, Slack, etc.
UI Design	Simple, lightweight, minimalistic	Modern, interactive, but heavier
Ease of Use	Easier to grasp but less polished	Steeper learning curve (many features)
Updates & Patches	Community updates (manual for self-hosted)	Maintained by Atlassian (automatic for Cloud)
Data Privacy	Full control if self-hosted	Managed by Atlassian (Cloud)

Conclusion

While Jira and Redmine both satisfy core requirements for multi-project issue tracking and collaboration, they reflect distinct design priorities. Jira's cloud-centric ecosystem emphasizes rapid time-to-value through native Agile tooling, rich automation (e.g., advanced querying), and a curated marketplace, alongside enterprise governance, identity, and compliance features, benefits that come with recurring subscription costs and platform dependence. Redmine, as a GPL-licensed, self-hosted solution, privileges infrastructural control, source-level extensibility, and zero license fees, but shifts responsibility for reliability, security hardening, upgrades, and feature coverage (often via plugins) to the adopting organization. Consequently, Jira tends to be the stronger choice where speed of rollout, vendor support, and standardized guardrails are paramount, whereas Redmine is well-suited to teams that value data sovereignty, deep customization, and cost discipline and possess sufficient DevOps capacity to manage the stack. The superior option is therefore contingent on context-specific constraints, including regulatory obligations, integration needs, total cost of ownership, and the maturity of internal operational capabilities.