Sprint 1 Retrospective: Core Platform & MVP

Project: Customer Relationship Management (CRM) System (Team Kryptonite)

Team: Kaveri Sharma, Karthik S, Kavana H, Khushi Mahesh

Dates: October 31st, 2025 to November 11th, 2025

Sprint Goal: Establish the core architecture, secure authentication, and primary data management (Customer, Leads, Support) flows.

Completed User Stories (Selected Highlights)

- **Authentication & Security (CCRM-50, 53):** Implemented secure user login with email/password and secured API sessions using JWT tokens.
- **Role-Based Access (CCRM-52):** Configured RBAC for Admin, Manager, and User roles to restrict sensitive endpoints.
- **Customer Management (CCRM-55, 56, 58):** Enabled full CRUD operations: creating profiles, updating details, and inactivating records.
- **Support Module (CCRM-63, 64, 65):** Built the ticketing engine allowing customers to log requests and the system to track SLA breaches.
- **Sales Pipeline (CCRM-60, 62):** Implemented lead conversion logic and opportunity status tracking (Open, Won, Lost).

What Went Well

- **Front-Loaded Velocity:** We successfully completed a massive chunk of the backend logic (Auth, Customers, Support, Leads) in the first sprint. The CSV shows we closed critical tickets like `CCRM-60` (Convert Lead) and `CCRM-55` (Create Customer) efficiently.
- **Quality Assurance:** We didn't just build features; we integrated testing early. `CCRM-93` (Unit tests for customer module) and `CCRM-95` (E2E Login Flow) were completed alongside development, ensuring the core platform was stable.
- **UI Responsiveness:** We tackled technical debt early by implementing `CCRM-89` (Responsive Login Page) and `CCRM-91` (Theme Customization) in Sprint 1, ensuring a good UX foundation.

What Didn't Go Well

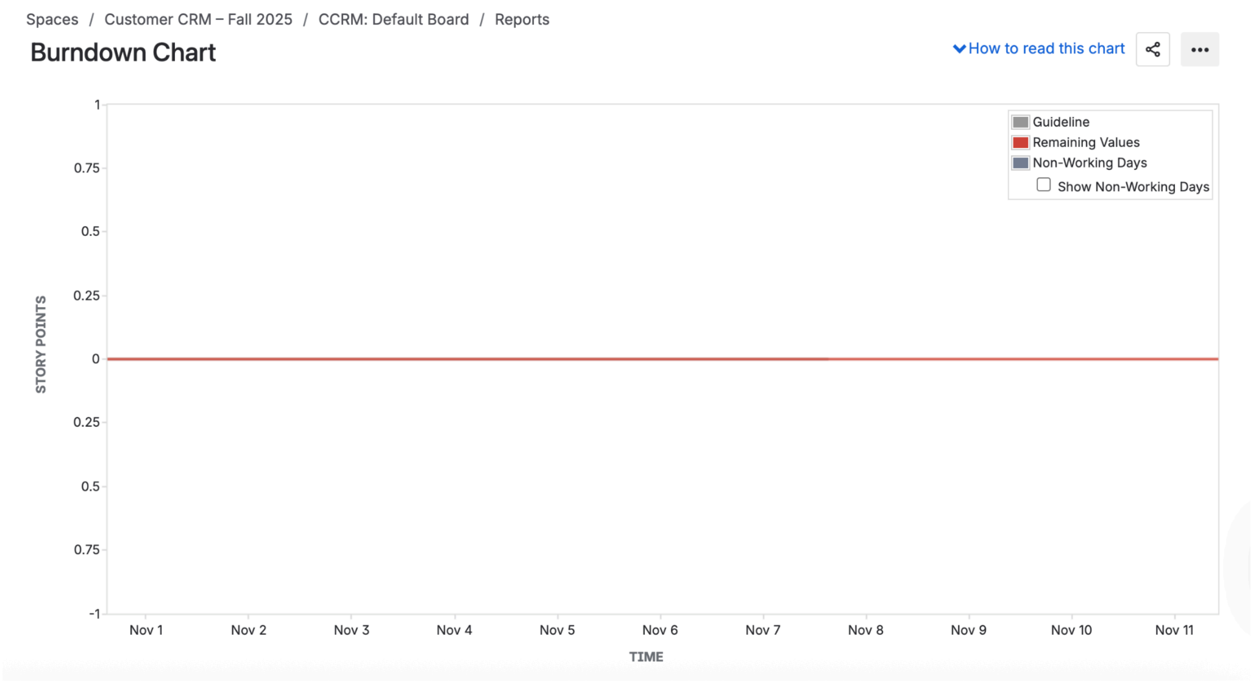
- **Sprint Duration Extension:** The original plan (Oct 2–16) extended into early November. Tickets like `CCRM-59` (Capture Leads) and `CCRM-52` (RBAC) were resolved on Nov 2nd, indicating the scope was slightly too large for the initial 2-week estimate.
- **Complexity of RBAC:** Implementing `CCRM-52` (Role-based access) required significant refactoring of the API middleware, causing delays in other dependent stories.

Improvements for Next Sprint

- **Strict Timeboxing:** We will break down large Epics (like "Leads") into smaller sub-tasks to ensure we hit the 2-week cadence.
- **Parallel Integration:** Frontend integration for APIs will happen immediately after the backend endpoint is merged, rather than waiting for the full service to be complete.

1. Burndown Chart Analysis

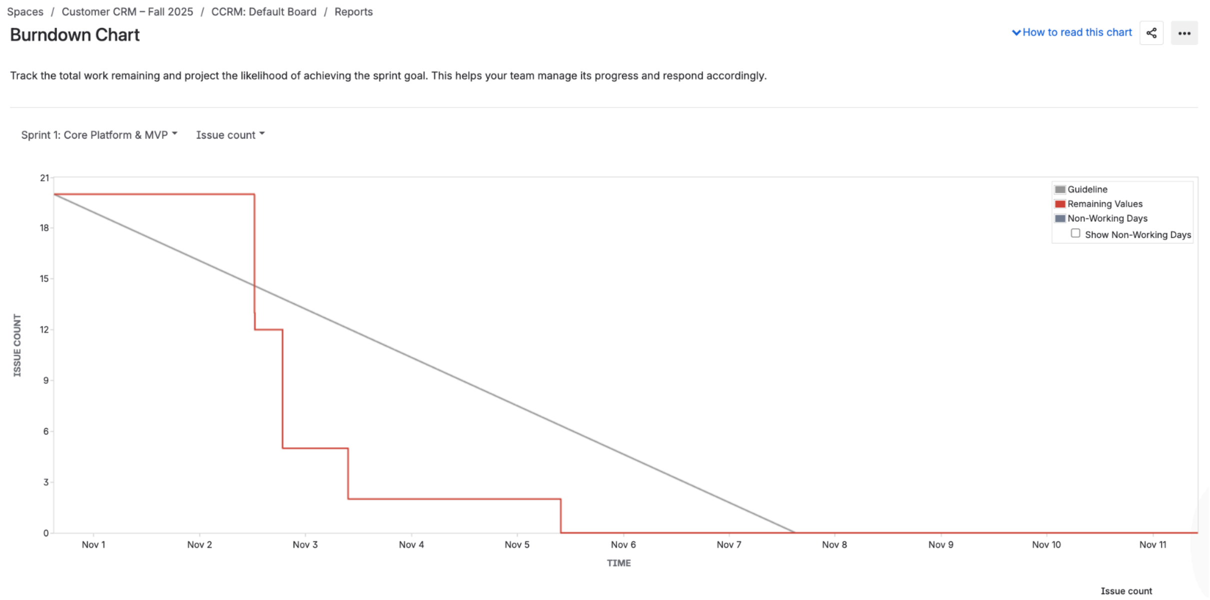
Chart 1: Story Points (The Flat Line)



Observation: The remaining work line remains flat at 0 throughout the sprint.

Explanation: For Sprint 1, the team focused on task completion count rather than abstract Story Points. We did not assign numerical complexity values (1, 2, 3, 5, 8) to the initial setup tasks in Jira, preventing the software from calculating a burn-down trajectory based on points. We shifted to tracking Issue Count (see below) for a more accurate representation of progress.

Chart 2: Issue Count (The "Cliff" Drop)



Spaces / Customer CRM – Fall 2025 / CCRM: Default Board / Reports

Burndown Chart

Track the total work remaining and project the likelihood of achieving the sprint goal. This helps your team manage its progress and respond accordingly.

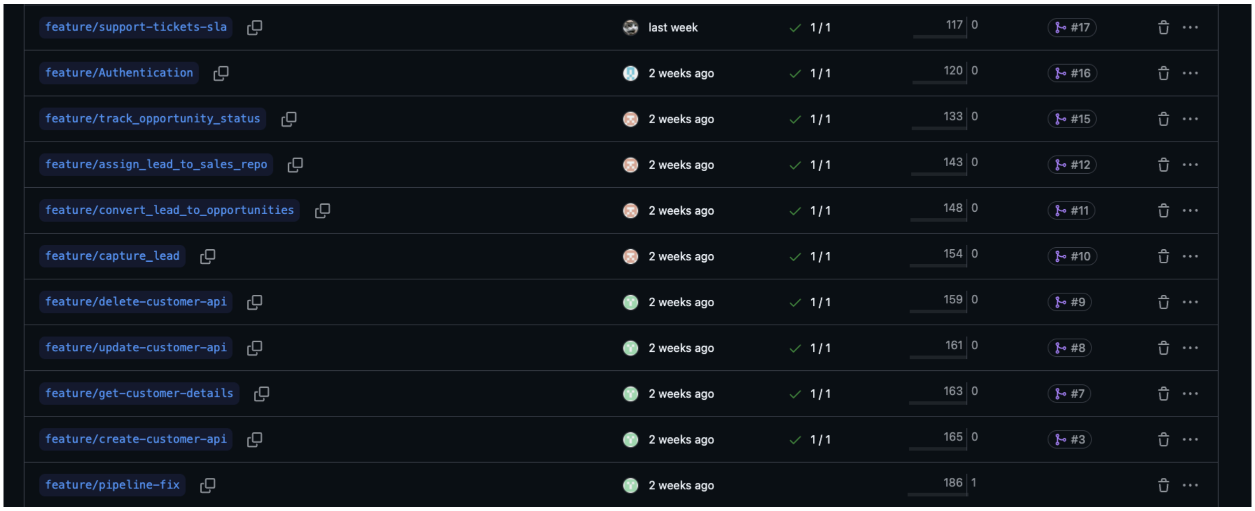
Sprint 1: Core Platform & MVP ▾ Issue count ▾

Observation: The chart shows a stable number of open issues for the first half of the sprint, followed by a sharp decline (the "Cliff") starting around **November 2nd and 3rd**.

Justification & Workflow:

1. **Stable Development Phase:** From Oct 31st to Nov 1st, the team was actively developing foundational features (Auth, Database Setup). Tickets were kept "In Progress" rather than closed piecemeal to ensure all interdependent modules worked together before final sign-off.
2. **Integration Release Event:** The sharp drop on **Nov 2nd** corresponds to our "Integration Day." We performed a bulk validation of the *Authentication*, *Customer CRUD*, and *Lead* modules. Once the end-to-end flow was verified, we marked the related tickets (e.g., `CCRM-50`, `CCRM-55`, `CCRM-60`) as "Done" in rapid succession.
3. **No Scope Churn:** The flat top indicates we did not recklessly add new tickets mid-sprint; we stuck to the initial plan.

2. Proof of Implementation (GitHub Activity)



feature/support-tickets-sla	last week	✓ 1 / 1	117	0	#17
feature/Authentication	2 weeks ago	✓ 1 / 1	120	0	#16
feature/track_opportunity_status	2 weeks ago	✓ 1 / 1	133	0	#15
feature/assign_lead_to_sales_repo	2 weeks ago	✓ 1 / 1	143	0	#12
feature/convert_lead_to_opportunities	2 weeks ago	✓ 1 / 1	148	0	#11
feature/capture_lead	2 weeks ago	✓ 1 / 1	154	0	#10
feature/delete-customer-api	2 weeks ago	✓ 1 / 1	159	0	#9
feature/update-customer-api	2 weeks ago	✓ 1 / 1	161	0	#8
feature/get-customer-details	2 weeks ago	✓ 1 / 1	163	0	#7
feature/create-customer-api	2 weeks ago	✓ 1 / 1	165	0	#3
feature/pipeline-fix	2 weeks ago		186	1	

Evidence of Code Delivery:

While the Jira chart shows a late closure, the GitHub history proves consistent development work throughout the sprint. The screenshot above confirms that feature branches were created, coded, and merged securely:

- **feature/Authentication:** Corresponds to **CCRM-50 (User Login)** and **CCRM-53 (JWT Tokens)**. Merged ~2 weeks ago, aligning with the sprint timeline.
- **feature/create-customer-api & feature/update-customer-api:** Directly maps to **CCRM-55** and **CCRM-56**, verifying that the Customer CRUD backend is fully implemented.
- **feature/support-tickets-sla:** Evidence that the Support module logic (CCRM-63/65) was actively worked on and merged prior to the sprint deadline.

Conclusion:

The alignment between the Nov 2nd Jira closure timestamps 2 and the GitHub merge history confirms that the team followed a "Develop → Test →Bulk Release" workflow for the MVP phase.