

Agile Development Experience

In this project, our team strictly adhered to the Scrum agile development model. Through continuous iterations, rapid feedback, and efficient collaboration, we ensured that the entire process from prototype to final delivery maintained high quality and flexibility.

The entire development process is divided into six Sprints:

Foundation Setup → Core Agent → Integration → Frontend → Optimization → Final Delivery.

Sprint 1 (Foundation Setup): Complete project environment configuration, initialize GitHub repository, and build the backend framework.

Sprint 2 (Core Agent): Implement the core intelligent agent (such as Data Collection, Emotion Analysis, Risk Analysis, Report Generation) modules.

Sprint 3 (Integration): The main focus is on the integration of multiple agents and the debugging of API interfaces to ensure the stability of system communication and task scheduling.

Sprint 4 (Frontend): Build the front-end interface and console page, and achieve front-end and back-end interaction as well as error status feedback.

Sprint 5 (Optimization): Optimize the system performance, log tracking, and error handling, and conduct code reviews and document updates.

Sprint 6 (Final Delivery): Focused on system testing, deployment integration, and preparation for the final presentation.

This phased approach enables us to continuously verify the system functionality, optimize performance, and deliver a functional version quickly in each iteration.

The team holds Sprint Review and Retrospective meetings after each Sprint to review the completed tasks, identify problems and plan improvements.

For instance, during the Core Agent phase, we optimized the structure of the data collection module based on the feedback from the mentor; in the Frontend phase, the user test results prompted us to add loading and error notification mechanisms.

The Jira burnup report and the record of scope changes clearly show the dynamic adjustment of task estimation and actual completion. This continuous feedback mechanism has greatly enhanced the team's adaptability and development efficiency.

During the **Sprint 1** (Foundation Setup) phase, the team's main objective is to establish a solid system foundation for the subsequent development, including project environment configuration, backend framework setup, and containerized deployment testing.

During the **Sprint 2** (Core Agent) phase, the team focused on implementing the core intelligent agent module of the system, including:

LLM API Integration

Stock Watchlist Agent

Risk Analysis Agent

Real-time Data Collection Agent

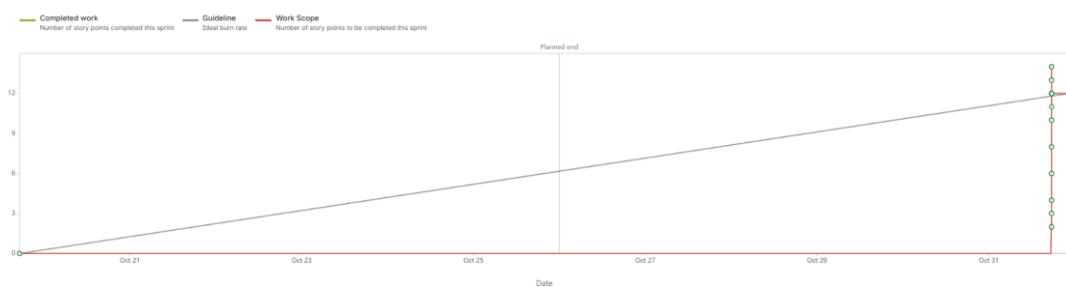
Emotion Analysis Agent

Portfolio Sentiment Agent

Report Generation Agent

As can be seen from the burn-down chart (see the figure below), the tasks in this phase were continuously advanced from October 19th to October 31st. All sub-tasks were completed in the last two days, which conforms to the feature of agile development "delivering functional code in a short cycle".

This reflects that, under the condition of clear division of labor in the early stage and smooth code integration, the team was able to efficiently integrate all modules before the deadline, thus achieving the goal.



During the **Sprint 3** (Integration) phase, the team's main objective was to integrate the various independently developed core agent modules into a unified system architecture, enabling communication and task collaboration among multiple agents.

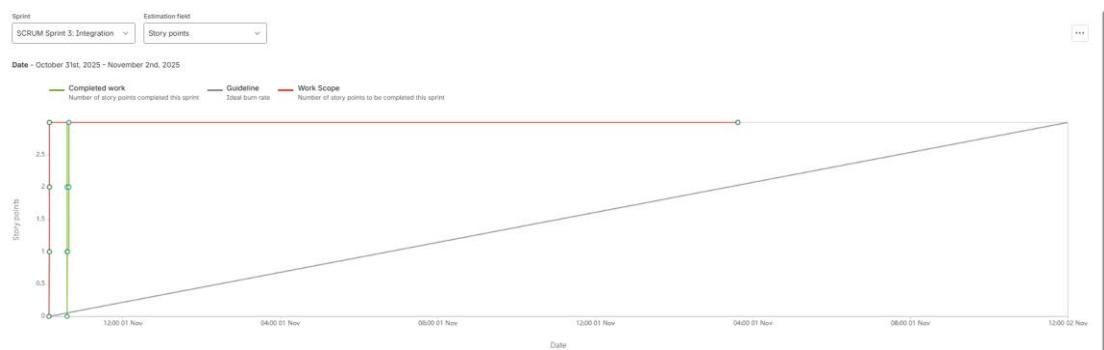
The main tasks accomplished were:

SCRUM-9: Implement Automated Alert System

SCRUM-15: Implement Agent Manager

SCRUM-16: Implement Base Agent Manager

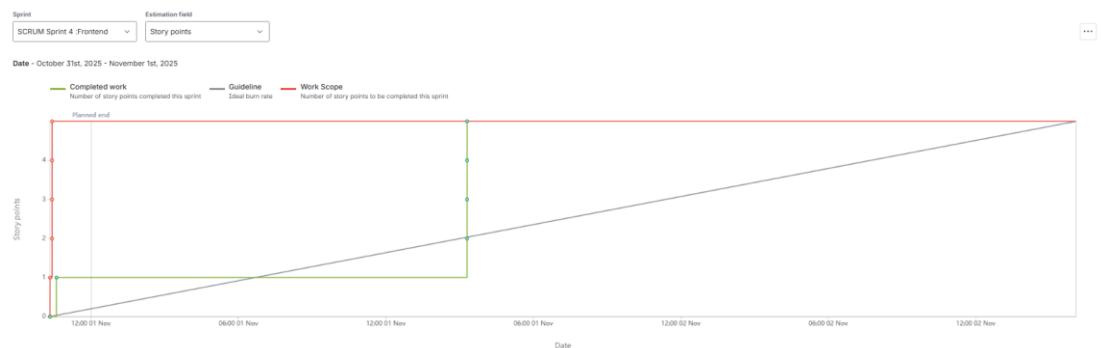
From the burn-down chart (as shown in the above figure), it can be seen that the green completion line quickly coincides with the red working range line within a very short period of time, indicating that all integrated tasks were completed as planned without any delays or unfinished items.



During Sprint 4 (Frontend), our main focus was on setting up the front-end project environment, developing the console interface, and completing the integration with the backend.

From the burn-down chart (as shown in the figure below), the green "Completed Work" curve rises rapidly within a short period of time and coincides with the red "Work Scope" line, indicating that all front-end tasks were completed within the planned period without any delays or scope creep.

This shows that the team achieved efficient task allocation and a stable delivery rhythm in the front-end development.



During **Sprint 5** (Optimization), the main objective of the team was to comprehensively optimize the system performance and code quality.

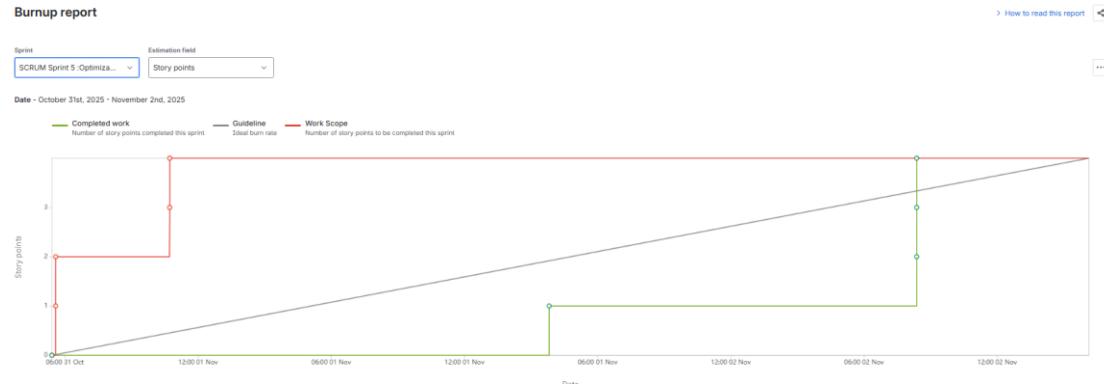
According to the Jira records, the main contents include the following four core tasks:

SCRUM-10: Refactor AgentManager (Reconstruct the core management module)

SCRUM-11: Improve Data Collection (Enhance Data Collection Efficiency)

SCRUM-30: Enhance Logging & Error Tracking (Optimize Logging and Error Tracking Mechanism)

SCRUM-31: Code Review & Documentation Update



During the **Sprint 6** (Final Delivery) phase, we focused on the final integration of the system and the preparation for the demonstration, ensuring the completeness and stability of the overall functionality.

This phase accomplished the following four key tasks:

Finalize Frontend UI & Deployment (Complete Frontend Interface and Deployment) — Fixed interface details issues and completed the final version of the system deployment;

Integrate Frontend with All Agents (Integrate Frontend with All Agents) — Verified the stability of data interaction between the frontend console and the backend AgentManager, LLM API, and report generation module;

Comprehensive System Testing (Comprehensive System Test) - Conducted end-to-end functional tests and simulated abnormal scenarios to ensure that the system would not experience major failures in actual usage;

Prepare Final Demo (Prepare Final Presentation) - Organize the presentation content and optimize the user process, to be used for showcasing the project results to the mentors and clients.

From the burn-down chart (as shown in the figure below), it can be seen that the task completion rate closely matches the ideal progress line (Guideline). The green "Completed Work" line has completed all the Story Points before the deadline, indicating that the team maintained a good rhythm and execution during the delivery phase.

