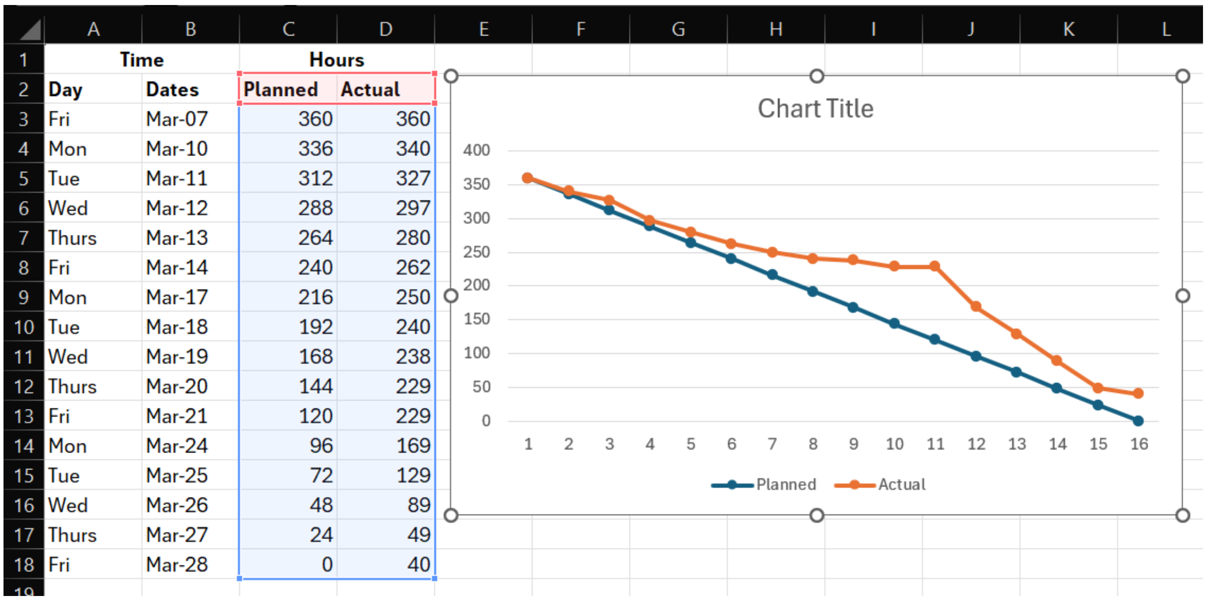
UFCFTR-30-3

Distribute & Enterprise Software Development

Sprint Review 2

Group:	11
Sprint:	2
Members:	Yamin Shwe Yi Htay, Vanessa Brown, Mohamad Abdou Amgad, Jack Harris, William Forber

Burn-down Chart

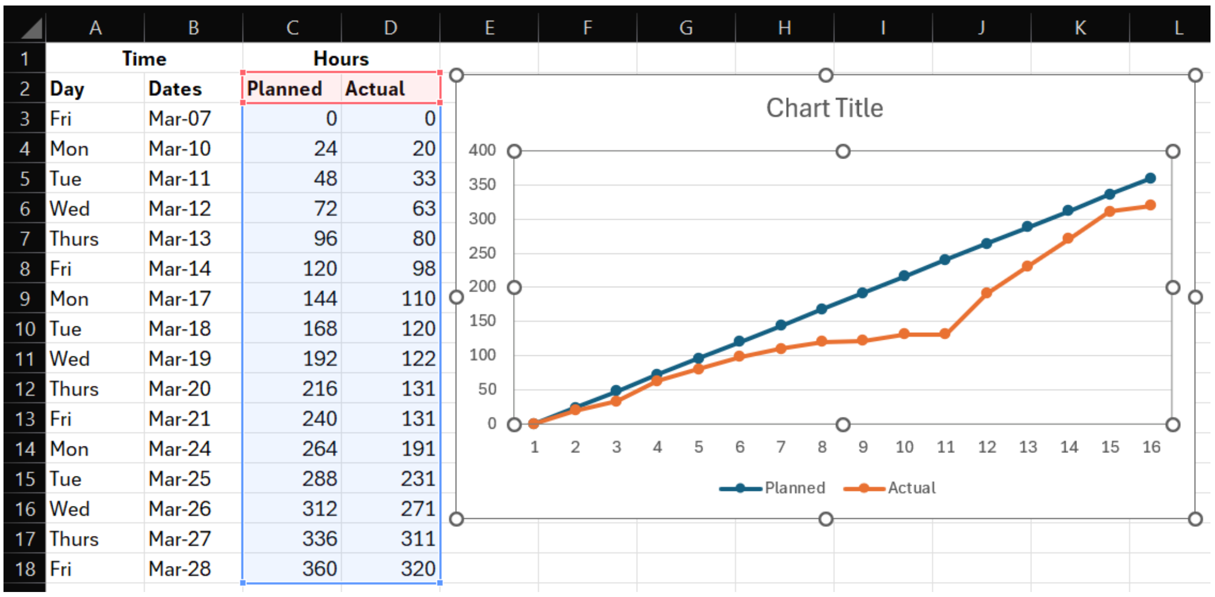


Time		Hours	
Day	Dates	Planned	Actual
Fri	Mar-07	360	360
Mon	Mar-10	336	340
Tue	Mar-11	312	327
Wed	Mar-12	288	297
Thurs	Mar-13	264	280
Fri	Mar-14	240	262
Mon	Mar-17	216	250
Tue	Mar-18	192	240
Wed	Mar-19	168	238
Thurs	Mar-20	144	229
Fri	Mar-21	120	229
Mon	Mar-24	96	169
Tue	Mar-25	72	129
Wed	Mar-26	48	89
Thurs	Mar-27	24	49
Fri	Mar-28	0	40

Figure 1 - Burn-down Chart (Hours)

Total hours were reduced from 384 (over 16 days) to 360 in order to account for planned hours being reduced to 0 on the final day and a clean calculation of whole numbers in the planned estimates. The actual progress deviated from the ideal burndown trajectory, reflecting delays in key areas such as frontend implementation and AI model integration. Instead of a steady decrease, progress stalled at certain points, particularly leading up to and on the 12th day of the sprint, where work remained at around 129 hours before proceeding to decrease at an adequate rate.

Burn-up Chart

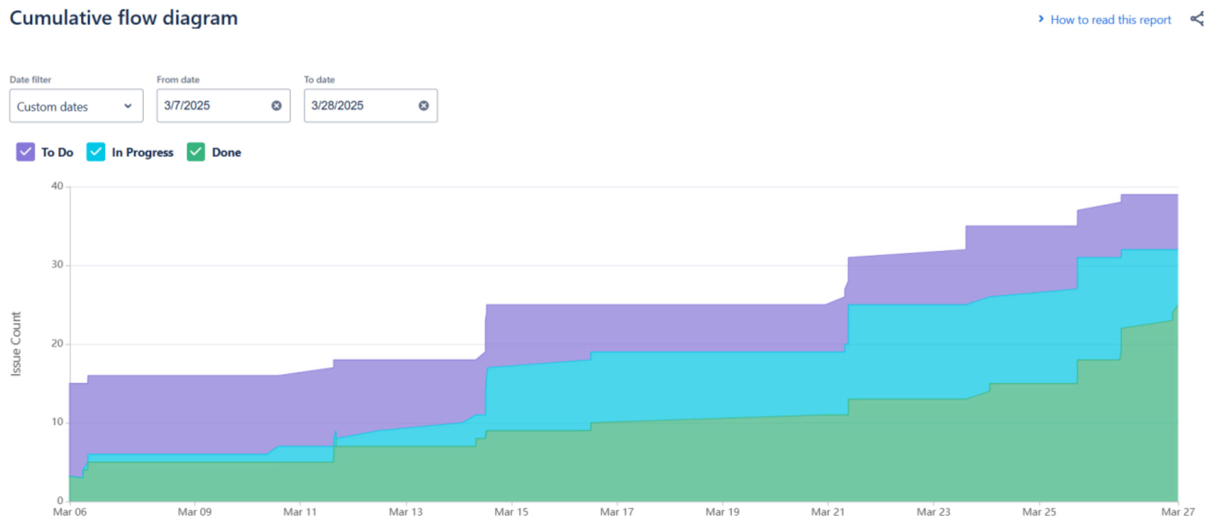


Day	Dates	Planned	Actual
Fri	Mar-07	0	0
Mon	Mar-10	24	20
Tue	Mar-11	48	33
Wed	Mar-12	72	63
Thurs	Mar-13	96	80
Fri	Mar-14	120	98
Mon	Mar-17	144	110
Tue	Mar-18	168	120
Wed	Mar-19	192	122
Thurs	Mar-20	216	131
Fri	Mar-21	240	131
Mon	Mar-24	264	191
Tue	Mar-25	288	231
Wed	Mar-26	312	271
Thurs	Mar-27	336	311
Fri	Mar-28	360	320

Figure 2 - Burn-up Chart (Hours)

The burn-up chart displays cumulative work completed over time. While we expected steady progress, deviations emerged due to unexpected challenges. Specifically, frontend and backend integration issues delayed task completion. The expected rate of completion was not achieved by the end of the 2nd week. However, by leveraging tools like charts, diagrams, and the SCRUM board, we identified these setbacks early and intensified efforts to successfully complete most of the remaining work.

Cumulative Flow Diagram



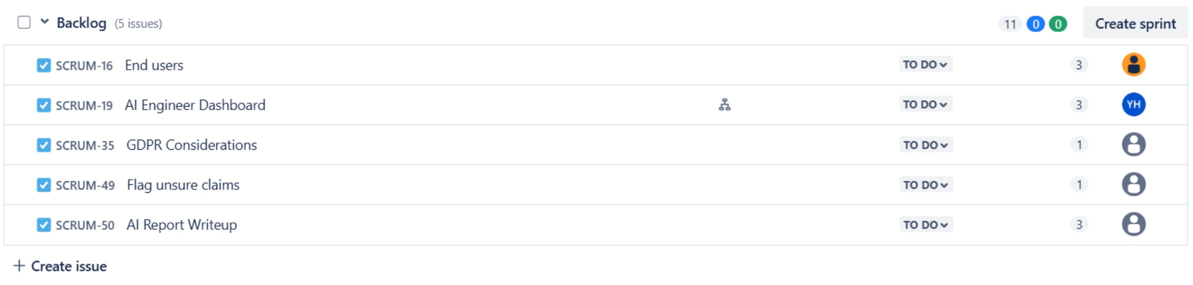
Cumulative flow diagram

Date filter: Custom dates
From date: 3/7/2025
To date: 3/28/2025

Figure 3 - Cumulative Flow Diagram (Sprint 2)

The cumulative flow diagram illustrates the distribution of work across different phases. As new tasks were pulled from the backlog, work-in-progress fluctuated due to bottlenecks in frontend implementation. While the supporter roles and review process helped mitigate the majority of issues, the sprint concluded with unfinished AI model comparison report tasks.

Backlog List



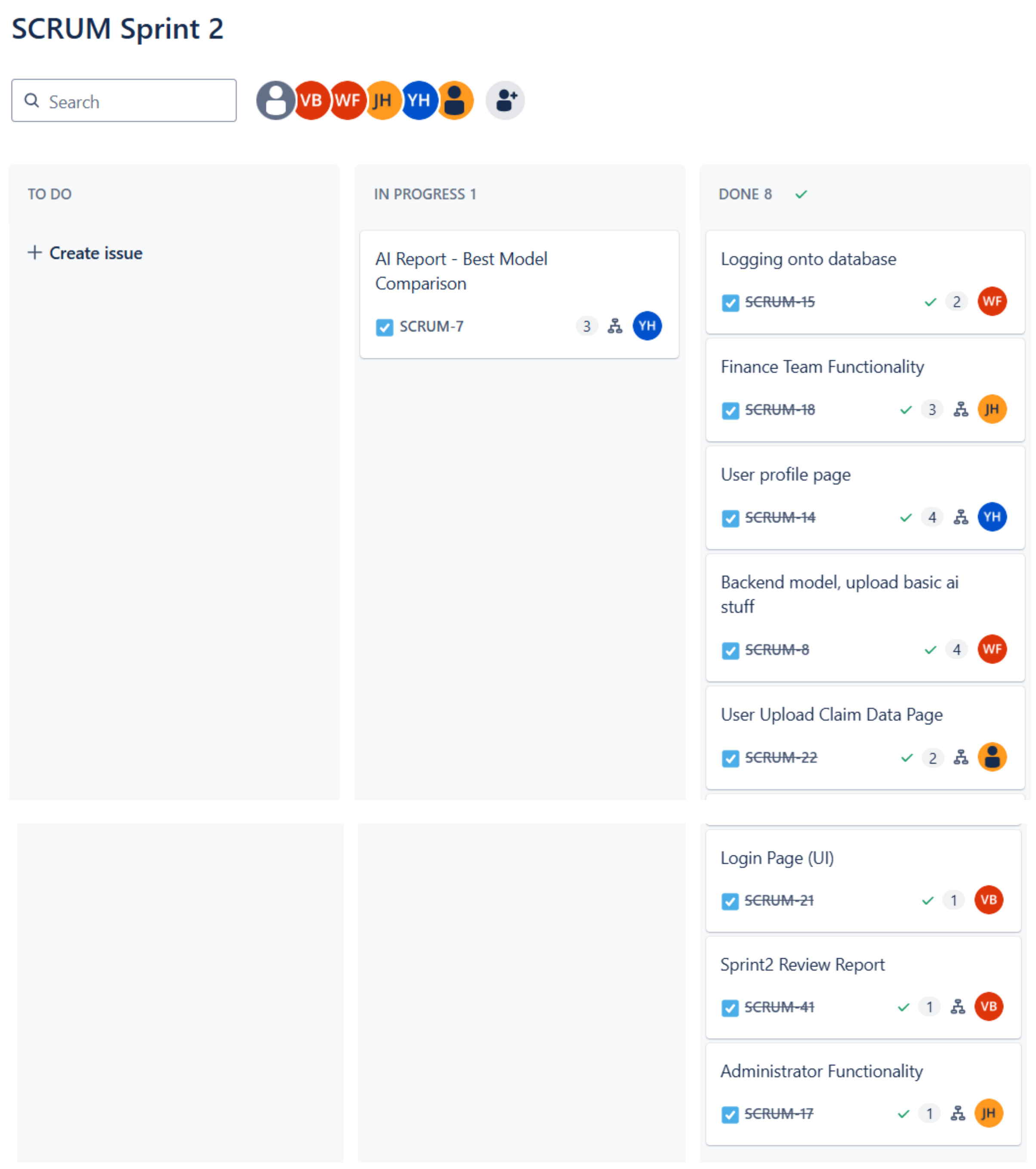
Backlog (5 issues) 11 0 0 Create sprint

SCRUM-16	End users	TO DO	3	
SCRUM-19	AI Engineer Dashboard	TO DO	3	YH
SCRUM-35	GDPR Considerations	TO DO	1	
SCRUM-49	Flag unsure claims	TO DO	1	
SCRUM-50	AI Report Writeup	TO DO	3	

+ Create issue

Figure 4 - Backlog (End of Sprint 2)

SCRUM Board


SCRUM Sprint 2

TO DO

+ Create issue

IN PROGRESS 1

AI Report - Best Model Comparison
SCRUM-7 3 YH

DONE 8

Logging onto database
~~SCRUM-15~~ 2 WF

Finance Team Functionality
~~SCRUM-18~~ 3 JH

User profile page
~~SCRUM-14~~ 4 YH

Backend model, upload basic ai stuff
~~SCRUM-8~~ 4 WF

User Upload Claim Data Page
~~SCRUM-22~~ 2

Login Page (UI)
~~SCRUM-21~~ 1 VB

Sprint2 Review Report
~~SCRUM-41~~ 1 VB

Administrator Functionality
~~SCRUM-17~~ 1 JH

Figure 5 - Scrum Board (Sprint 2)

The SCRUM board was structured based on priority, but some adjustments were necessary during the sprint. Some tasks were deprioritized, while others required additional iterations to meet requirements.

Final Status of Tasks

Completed:

- Logging onto database
- Finance Team Functionality
- User profile page
- Backend model setup, basic AI functionality
- User Upload Claim Data Page
- Login Page (UI)
- Sprint2 Review Report
- Administrator Functionality

In Progress:

- AI Report – Best Model Comparison

Communication Issues and Challenges

One of the main challenges encountered during the sprint was the misalignment between the frontend and backend teams. There was initial confusion regarding whether the frontend should rely on Django templates or adopt a JWT authentication approach. This uncertainty led to incorrect references in the authentication flow, causing failed authentication attempts and template rendering issues. Additionally, the team was not fully aligned on whether to continue using Django templates or transition to React, which resulted in the creation of unnecessary React code that was ultimately not integrated. Further complicating matters, some frontend components attempted to create new models instead of leveraging existing backend endpoints, leading to inconsistencies in data handling and workflow.

Branching and code management also presented challenges. At one point during development, some implementations were inadvertently pushed directly to the main branch instead of being handled within feature branches. This not only bypassed the review process but also introduced bugs into the main codebase, leading to confusion and additional time spent troubleshooting and resolving issues. The lack of strict adherence to branching policies slowed progress, particularly in cases where unexpected changes disrupted existing functionality. To address these issues in the next sprint, we discussed implementing a token-based system to better manage merges and prevent conflicts. Under this approach, only the person holding the token would be allowed to merge into the main branch. Before merging, they would first pull the latest changes, update their feature branch, and ensure compatibility before proceeding. Once the merge is complete, the token would then be made available for the next developer. By enforcing a structured merging process, we aim to reduce conflicts, maintain code stability, and improve overall workflow efficiency.

Despite these challenges, the practice of assigning supporters and reviewers to tasks proved to be a valuable safeguard. Thanks to effective coordination and proactive issue tracking, most of these problems were identified and addressed before they could significantly impact the sprint. The support roles played a crucial part in debugging, reviewing API interactions, and ensuring dependencies were properly managed. However, moving forward, clearer guidelines on frontend/backend responsibilities and stricter enforcement of branching policies could further improve efficiency and prevent similar setbacks in future sprints.

Reflections

The burndown and burn-up charts illustrate that the total work completed did not fully align with initial expectations. Delays arose due to coordination challenges, particularly between the frontend and backend teams, as well as the need for multiple iterations on certain features to ensure they met functional requirements. While the burndown chart suggests a loss of momentum towards the end of the sprint, the cumulative flow diagram provides a more nuanced perspective. It indicates that our initial pace was on track, but prematurely introducing additional tasks mid-sprint disrupted our workflow. These added tasks, combined with existing integration challenges, temporarily slowed progress. However, despite these obstacles, we still managed to achieve significant advancements, demonstrating adaptability and perseverance in overcoming setbacks.

We maintained a strong collaborative dynamic throughout the sprint, holding regular meetings to align on priorities, address roadblocks, and ensure that everyone remained on the same page. Our continued practice of leadership rotation fostered accountability, allowing each team member to take ownership of different aspects of the project while ensuring a well-rounded distribution of responsibilities. Every team member played a crucial role, contributing not only through development efforts but also by supporting others—whether by reviewing code, troubleshooting issues, or refining our implementation approach. Despite the challenges we encountered, our ability to work cohesively, adapt to obstacles, and push forward with determination highlights the resilience and commitment of the team.

Moving forward, a more structured approach to task management may help mitigate these challenges. Limiting the number of tasks pulled from the backlog while there are still unresolved in-progress tasks could prevent workload fragmentation. This would ensure that if a review uncovers issues, team members have the bandwidth to address them promptly, rather than shifting focus to lower-priority tasks. By enforcing clearer task prioritization and review cycles, we can improve overall sprint efficiency and reduce the risk of unfinished work carrying over into future sprints.

Key Takeaways

- Better frontend-backend coordination is needed to avoid redundant implementations and integration delays.
- Stricter branching policies should be enforced to prevent issues caused by direct merges into main.
- More structured task dependencies would help in managing unfinished work.
- The practice of leadership rotation and regular team meetings proved valuable in maintaining accountability and alignment—continuing this approach in the next sprint will help sustain team cohesion and efficiency.

Final Thoughts

This sprint highlighted the importance of structured collaboration, the need for better branch management and improved communication between frontend and backend teams. Refining our integration testing approach and ensuring a smoother transition from planning to execution will be key to improving overall sprint success.

Despite the challenges encountered, strong development practices and a solid early sprint performance meant that we maintained overall progress. While certain issues caused delays, the team remained proactive in identifying and addressing problems. Moving forward, by reinforcing clearer workflows and prioritizing effective coordination, we can build on this momentum and continue driving the project forward.

Relevant Links

GitLab Repo Link -> https://gitlab.uwe.ac.uk/y2-htay/insurance-claim-predictor

Jira Project Link -> https://insurance-project.atlassian.net/jira/software/projects/SCRUM/summary