

Assignment 2 - Project Management Plan

Cloud Carbon Sense

Team Name / Team Number: GoGreen / 1311

Student Name and ID:

Cong Duc Nguyen - 35480025

Tsz Ching James Chiang - 35719176

Rongze Ma - 35286288

Bhavesh Singh - 35103256

Date of Submission:

03/11/2025

Unit Code & Semester: FIT5057 – Semester 2, 2025

Table of Contents

1. Vision and Foundations	3
1.1. Project Vision	3
1.2. Problem Statement	3
1.3. Project Requirements	3
2. User-Centred Design	4
2.1. Personas	4
2.2. Journey Map for Key Persona	6
3. Agile Planning	7
3.1. Epics & User Stories	7
3.2. Sprint Allocation	15
3.3. Release Phase 1	17
3.4. Sprint 1 Execution Plan	18
4. Reflection	20
5. References	23
6. GenAI Declaration	24

1. Vision and Foundations

1.1. Project Vision

The Cloud Carbon Sense project will provide a clear, user-centric dashboard that helps to visualize, understand, and manage Monash's cloud-based carbon footprint. The system will provide actionable insights by interpreting complex cloud data, helping IT teams optimize cloud resources, and helping department heads track their progress. By transforming complex cloud data into clear, actionable insights for every department, this system will directly support Monash's strategic plan to address the global challenge of climate change (Monash University, 2021, p. 6) and its "effective commitment to the United Nations Sustainable Development Goals" (Monash University, 2021, p. 26).

1.2. Problem Statement

Currently, Monash's IT, finance, and research departments operate with invisible and unmanaged cloud carbon emissions. This creates a significant disconnect between technology usage and environmental responsibility, leading directly to wasted budgets on inefficient, high-carbon resources. The critical impact of this disconnect is a systemic inability to track or report on institutional sustainability targets. Our project, Cloud Carbon Sense, will resolve this by providing the real-time monitoring needed to make these emissions visible and manageable.

1.3. Project Requirements

To address this problem, we have defined six key requirements linked directly to our user needs and project goals:

Functional Requirements:

- The system must collect real-time carbon data from AWS, GCP, and Azure Cloud.
- Users must be able to view CO₂e (carbon equivalent) visualised by department, time range, and cloud platform.
- The system must propose at least one recommended action to reduce carbon emissions for the IT department.
- The system must allow users to generate and download a PDF summarising CO₂e and associated cost for a selected department and time range.

Non-Functional Requirements:

- To maintain user productivity and prevent task-disrupting context switches (Nielsen, 2010), the system must load simple informational pages in under 3 seconds (Think with Google, 2016), complex data-intensive operations (report generation) must be completed in under 8 seconds.
- To provide a highly reliable service for all university stakeholders, the system will target 99.9% availability (AXELOS, 2019), max 44 minutes of unplanned downtime per month.
- The system must comply with all Monash University data privacy and security standards.
- Constrained to an Azure-first stack and read-only API access, this project excludes on-prem data centre telemetry and per-user carbon accounting.

2. User-Centred Design

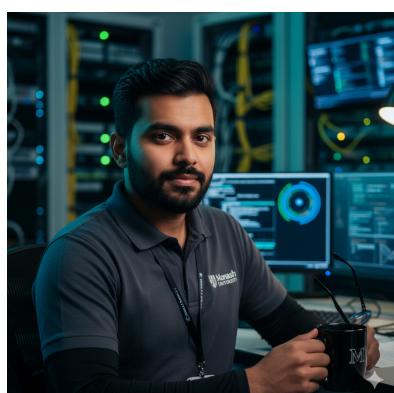
2.1. Personas

2.1.1. Prof. Todd Boehly (Department Head)



- **Role:** IT Faculty Head
- **Quote:** "I don't care how it works, just show me our carbon number for the quarter."
- **Background:** Male, 58, long-time academic, not highly technical. Under pressure from the university to report on sustainability KPIs.
- **Goals:** Confidently report his department's carbon footprint numbers upwards.
- **Frustrations:** Carbon contribution is a "black box." He's asked to report on sustainability KPIs, but data is manually compiled, conflicts with financial reports, and is too slow to answer simple questions like "Why did our carbon footprint jump 20% last month?"
- **Behavioural Insight:** He needs a single, trusted "Carbon KPI" on a dashboard.
- **Requirement shaping:** His need for quick answers shaped Functional Requirement 2: Users must be able to view CO₂e visualised by department, time range, and cloud platform.

2.1.2. John Terry (Infrastructure Manager)



- **Role:** Cloud Infrastructure Manager - ESolution
- **Quote:** It's not just about cost sprawl anymore. Now I've got people spinning up servers in the wrong region, and our carbon bill is exploding.
- **Background:** Male, 34, technical, security-focused. Manages Monash's Clouds.
- **Goals:** Maintain control. Identify and reduce resource waste (both cost and carbon).
- **Frustrations:** No visibility into who is responsible for emissions. He fights "shadow IT" and "carbon sprawl" - a new metric he can't control. He knows researchers use unoptimized, high-carbon servers, but has no tool to prove it or enforce policy.
- **Behavioural Insight:** He is motivated by efficiency. He will be the power user of the "recommendations" feature.
- **Requirement shaping:** His technical role and security-first mindset created the business case for Functional Requirement 3 (recommended action) and made Non-Functional Requirement 3 (Data Privacy and Security) non-negotiable.

2.1.3. Dr. Chen (Researcher)



- **Role:** Post-doctoral Research Fellow (Bioinformatics)
- **Quote:** My grant deadline is next week. I'll run my model on 1,000 cores if I have to. Should I care about carbon footprint?
- **Background:** Female, 29, Python/R, grant-funded. Runs large data models.
- **Goals:** Get research done fast. Easy access to powerful computing.
- **Frustrations:** Sees sustainability rules slow her research. She is frustrated by IT approvals, compute limits, and any tool that isn't fast. She is unaware of her carbon impact and doesn't understand why it's her problem.

- **Behavioural Insight:** She is indifferent to carbon impact; her motivation (speed) is a cause of the problem. The tool must educate her.
- **Requirement shaping:** Her need for speed and multi-cloud access validated Non-Functional Requirement 1 (load time) and highlighted the importance of Functional Requirement 1 (collecting data from cloud providers)

2.1.4. Sarah (Finance Manager)

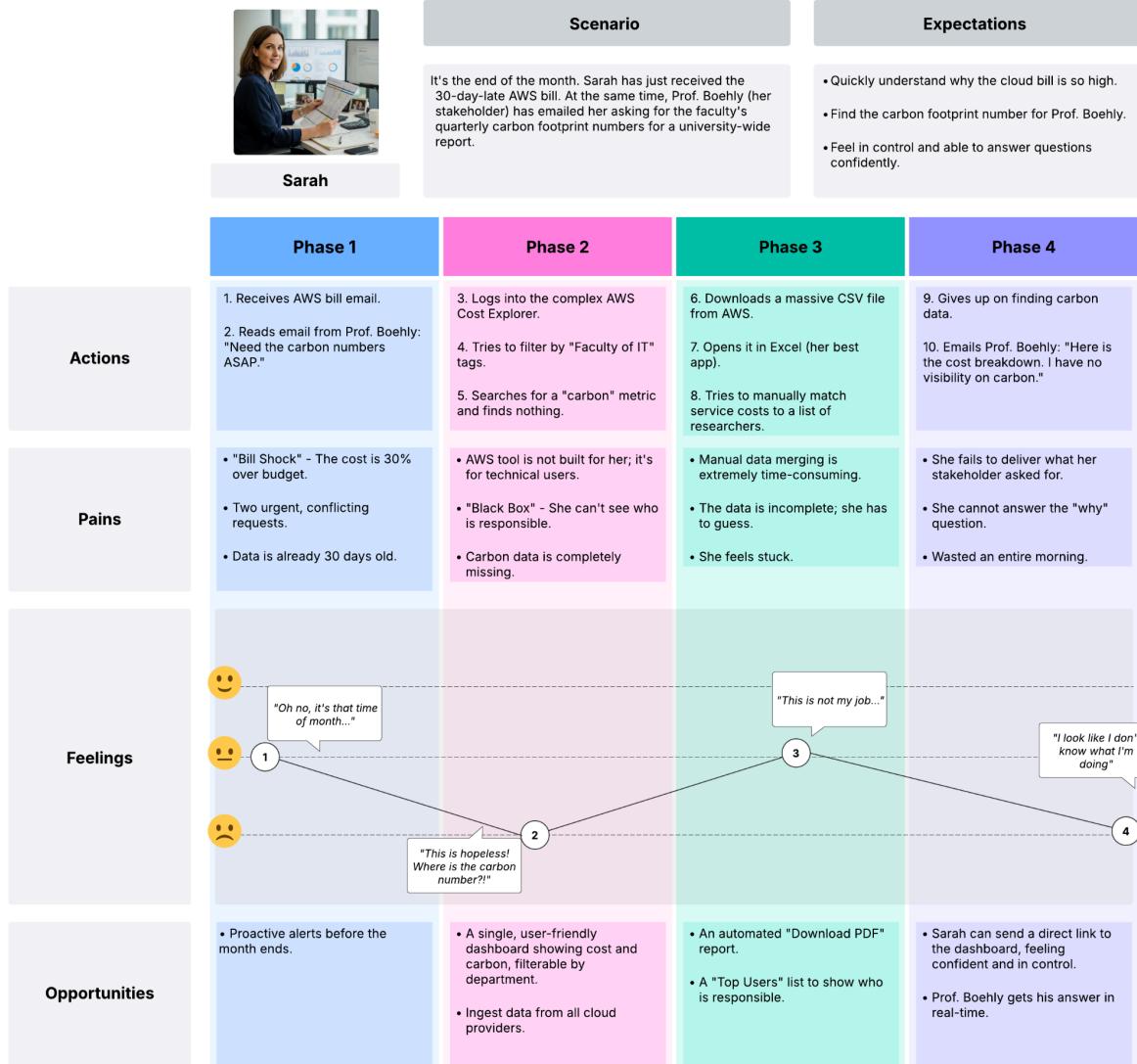


- **Role:** Head of Finance Department
- **Quote:** First, it was 'bill shock' for our costs. Now, I'm being asked to report on our carbon spend, and I have no idea where to even find that number.
- **Background:** Female, 45, Excel expert, non-technical. Monitors budgets.
- **Goals:** Predict periodic spend (dollars and carbon).
- **Frustrations:** Experiencing "bill shock" for both cost and carbon. She cannot link a specific dollar cost to a specific carbon cost, see which researcher is responsible, or forecast next month's carbon spend.

- **Behavioural Insight:** She needs to see cost and carbon side-by-side to understand the relationship.
- **Requirement shaping:** As a non-technical user who must provide reports to stakeholders directly created our reporting requirement: The system must allow users to generate and download a PDF report summarising carbon emissions and associated costs.

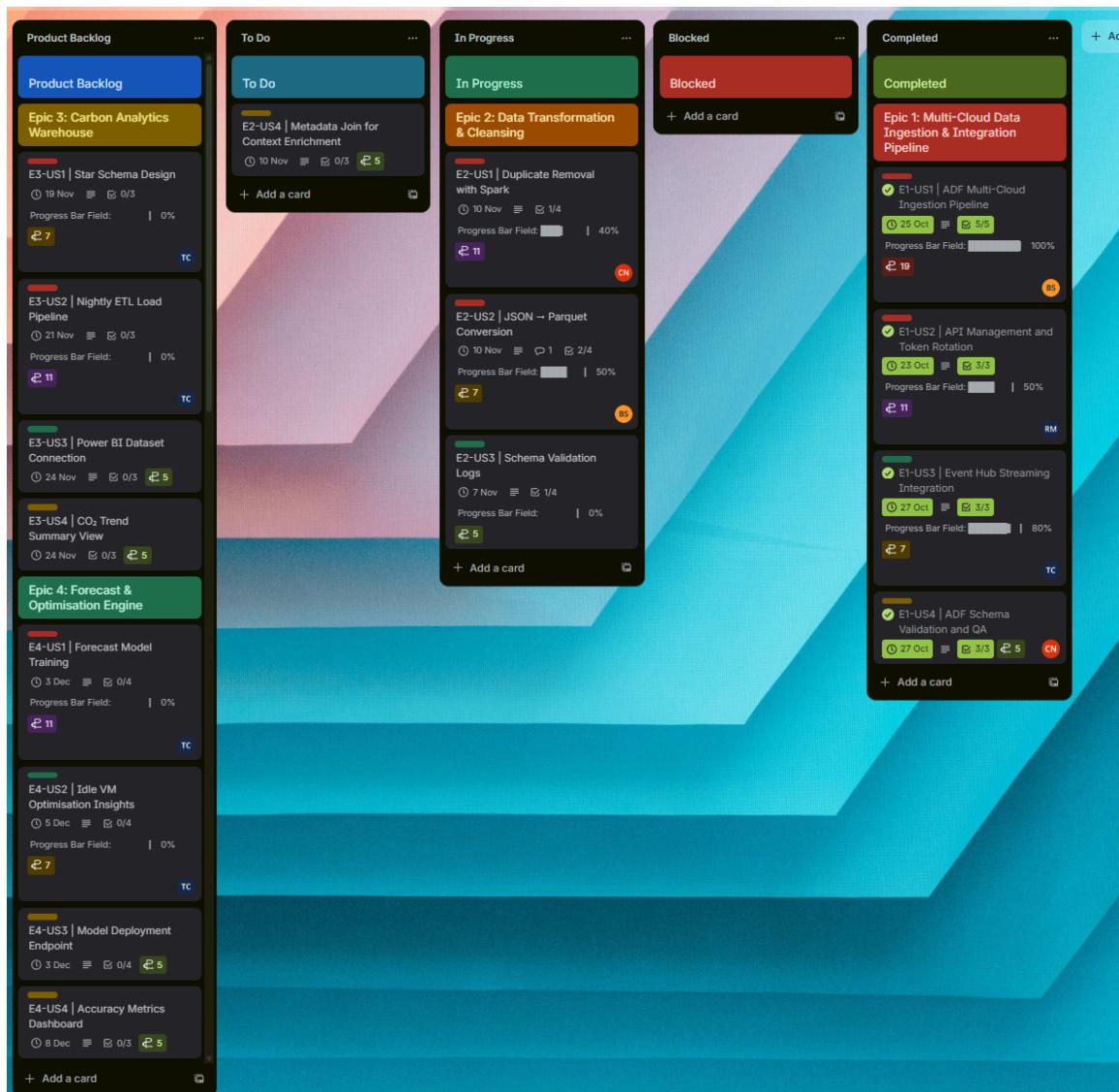
2.2. Journey Map for Key Persona

The following journey map details the current "as-is" experience, key pain points, and emotional journey of our key persona, Sarah (Finance Manager).



3. Agile Planning

3.1. Epics & User Stories



Epic-1 Multi-Cloud Data Ingestion & Integration Pipeline

Build reliable ingestion from AWS, Azure and GCP into ADLS Gen 2 using ADF, Event Hub and API Management.

E1-US1 | ADF Multi-Cloud Ingestion Pipeline

Completed

+ Add Checklist Attachment

Members	Labels	Due date	Story Points
BS	Must	25 Oct, 17:17 Complete	19

Description

As a Data Engineer, I want ADF pipelines to ingest CO₂ data from AWS, Azure and GCP so that metrics are centralised in ADLS for downstream analysis.

Acceptance Criteria:

- ADF pipelines securely connect to AWS, Azure and GCP APIs.
- Data lands in ADLS Gen 2 Bronze zone within 5 min of trigger.
- Authentication tokens rotate every 7 days.
- Pipeline logs include record count and latency metrics.
- Failed runs auto-retry and generate email alert.

Amazing Fields

Amazing Fields needs permission to access card data on this board. We use the minimum permissions needed to provide access and support. You can cancel authorization at any time.

For more details see: [Amazing Fields Permissions](#)

Authorize with Trello

DoD

Hide checked items Delete

100%

- Pipelines parameterised (resource ID, region).
- Schema and duplicate validation scripts executed.
- Run logs stored in monitoring table.
- Unit + integration tests ≥ 90 % coverage.
- Peer review completed and merged to Git main.

E1-US2 | API Management and Token Rotation

Completed

+ Add Checklist Attachment

Members	Labels	Due date	Story Points
RM	Must	23 Oct, 00:56 Complete	11

Description

As a Cloud Architect, I want API Management with token rotation so that data access is secure and auditable.

Acceptance Criteria:

- API gateway enforces JWT authentication and rotation ≤ 7 days.
- Access logs record all API calls with timestamps and user IDs.
- Unauthorized calls return HTTP 403 and alert security group.
- Logs stored in Key Vault-linked workspace for audit.

Amazing Fields

Amazing Fields needs permission to access card data on this board. We use the minimum permissions needed to provide access and support. You can cancel authorization at any time.

For more details see: [Amazing Fields Permissions](#)

Authorize with Trello

Definition of Done

Hide checked items Delete

100%

- Token rotation automation verified.
- Audit dashboard populated in Azure Monitor.
- Security review approved.

ID	User Story	MoSCoW	Sprint	Acceptance Criteria
E1-US1	As a Data Engineer, I want ADF to ingest AWS/Azure/GCP CO ₂ so that data is centralised in ADLS.	Must	1	Auth OK; load <5min; log summary.
E1-US2	As a Cloud Architect, I want API Management with token rotation so that access is secure and auditable.	Must	1	Token ≤7days; audit log
E1-US3	As a Data Engineer, I want Event Hub streaming so that dashboards update quickly.	Should	1	95% <2min.

E1-US4	As a QA Analyst, I want ADF schema checks so that mismatches are flagged.	Could	6	Daily report; ≤2% error.
--------	---	-------	---	--------------------------

Epic-2 Data Transformation & Cleansing

Transform raw Bronze data into curated Silver tables via Synapse Spark and Data Flows to ensure consistency and quality for analytics.

The image shows two Trello cards side-by-side, both titled "In Progress".

E2-US1 | Duplicate Removal with Spark

- Members:** CN
- Labels:** Must
- Due date:** 10 Nov, 22:39
- Story Points:** 11

Description:
As a Data Engineer, I want Spark jobs to remove duplicate records so that datasets remain accurate for analytics.

Acceptance Criteria:

- Spark job identifies and removes duplicates based on resource ID and timestamp.
- Duplicate rate ≤ 1 % after processing.
- Deduplication log summarises record count before/after cleaning.
- Job completes within 15 minutes for 100 GB dataset.
- Daily scheduled trigger runs successfully via Synapse notebook.

Amazing Fields:
Amazing Fields needs permission to access card data on this board. We use the minimum permissions needed to provide access and support. You can cancel authorization at any time.
For more details see: [Amazing Fields Permissions](#)

Visible Fields:
[Authorize with Trello](#)

Definition of Done:

- Spark code peer reviewed and merged to main branch.
- QA validated duplicate counts and log outputs.
- Unit tests cover ≥ 90 % of functions.
- Run history visible in Azure Monitor logs.

Progress: 25%

E2-US3 | Schema Validation Logs

- Members:** Should
- Labels:**
- Due date:** 7 Nov, 00:56
- Story Points:** 5

Description:
As a QA Analyst, I want schema validation logs so that transformation issues can be tracked and resolved quickly.

Acceptance Criteria:

- Validation runs after each load to compare source and target schemas.
- Daily summary generated and stored in "validation_logs".
- Mismatched columns highlighted with timestamp.
- Error rate ≤ 2 %.
- Notification sent to Slack/Teams on failure.

Amazing Fields:
Amazing Fields needs permission to access card data on this board. We use the minimum permissions needed to provide access and support. You can cancel authorization at any time.
For more details see: [Amazing Fields Permissions](#)

Visible Fields:
[Authorize with Trello](#)

Definition of Done:

- QA validated log structure and alert delivery.
- Automated pipeline tested on three sources.
- Unit coverage ≥ 85 %.
- Documentation updated in Confluence.

Progress: 25%

E2-US1	As a Data Engineer, I want Spark de-duplication so that datasets stay clean.	Must	2	Dup <1%.
E2-US2	As a Data Engineer, I want JSON → Parquet so that queries run faster.	Must	1	3× faster.

E2-US3	As a QA Analyst, I want schema logs so that errors are traceable.	Should	2	Table updates daily.
E2-US4	As a Data Analyst, I want metadata joined to region/service so that context is added.	Could	3	All rows tagged.

Epic-3 Carbon Analytics Warehouse

Develop a Synapse Dedicated SQL Pool to store aggregated emission metrics with a star schema and Power BI dataset for reporting.

The image shows two cards from a product backlog:

- E3-US2 | Nightly ETL Load Pipeline** (Must, 11 story points):
 - Description: As a Data Engineer, I want a nightly ETL job so that warehouse data stays current.
 - Acceptance Criteria:
 - ETL success \geq 98 % across all datasets.
 - Pipeline runs < 30 minutes for 100 GB load.
 - Job handles late arriving records and upserts.
 - Failure alerts triggered to Teams and email.
 - Log table records start/end time and row count.
 - Fields section: Amazing Fields, Visible Fields, Definition of Done (0% complete, Pipeline validated in Dev and Prod., Monitoring enabled in Synapse., Test coverage > 90%).
- E3-US3 | Power BI Dataset Connection** (Should, 5 story points):
 - Description: As a BI Developer, I want Power BI datasets connected to Synapse so dashboards stay up-to-date.
 - Acceptance Criteria:
 - Dataset auto-refreshes \leq 60 minutes.
 - Access secured with service principal.
 - No manual refresh required.
 - Connection validated for three Power BI reports.
 - Permissions approved by Data Owner.
 - Fields section: Amazing Fields, Visible Fields, DoD (0% complete, BI workspace approved by team., Dataset refresh test log attached to ticket., QA review passed.).

E3-US1	As a Data Modeller, I want fact/dimension tables so that KPIs are queryable.	Must	2	Star schema validated.
E3-US2	As a Data Engineer, I want nightly ETL into DW so that data stays current.	Must	2	ETL success rate \geq 98%.

E3-US3	As a BI Dev, I want the Power BI dataset connected to Synapse.	Should	3	Refresh ≤60min.
E3-US4	As a Data Analyst, I want CO ₂ trend view so that dept changes are visible.	Could	3	Query <2s.

Epic-4 Forecast & Optimisation Engine

Use Azure Databricks and ML models to predict future emissions and suggest resource optimisations.

The image shows two side-by-side screenshots of a Product Backlog interface, likely from a tool like Jira or Trello. Both screens have a header "Product Backlog" and a title for each story.

Left Screen (Story E4-US1):

- Title:** E4-US1 | Forecast Model Training
- Labels:** Must
- Due date:** 3 Dec, 22:39
- Story Points:** 11
- Description:** As a Data Scientist, I want to train an ML model so that CO₂ trends can be predicted for future months.
- Acceptance Criteria:**
 - Historical 12-month dataset loaded from Gold layer.
 - Model trained and versioned in MLflow.
 - RMSE ≤ 10 % on test set.
 - Feature importance table exported to Synapse.
 - Training log and metrics saved to Blob Storage.
- Fields:** Amazing Fields (Visible Fields)
- Definition of Done:**
 - Training notebook peer-reviewed.
 - Model artifact stored in registry.
 - Unit and integration tests ≥ 90 %.
 - Performance benchmark documented in Confluence.

Right Screen (Story E4-US3):

- Title:** E4-US3 | Model Deployment Endpoint
- Labels:** Could
- Due date:** 3 Dec, 22:39
- Story Points:** 5
- Description:** As a DevOps Engineer, I want to deploy the trained model as an API endpoint so that applications can fetch forecasts in real time.
- Acceptance Criteria:**
 - Endpoint deployed as Azure Function in Production.
 - Response time < 500 ms for 100 concurrent requests.
 - Error handling returns HTTP codes 200/400/500.
 - Uptime ≥ 99 %.
 - Health check endpoint implemented.
- Fields:** Amazing Fields (Visible Fields)
- DoD:**
 - Postman tests passed for all methods.
 - Logging enabled in App Insights.
 - Deployment approved by QA lead.
 - Release notes updated.

E4-US1	As a Data Scientist, I want a forecast model so that CO ₂ trends are predicted.	Must	3	RMSE <10%.
E4-US2	As an ML Engineer, I want optimization recommendations for idle VMs.	Should	4	≥5% saving.
E4-US3	As a DevOps Engineer, I want to deploy the model endpoint.	Could	6	Response <500ms.

E4-US4	As a Data Scientist, I want accuracy metrics logged.	Could	4	RMSE/F1 dashboard.
--------	--	-------	---	--------------------

Epic-5 Reporting - Visualisation & Dashboard UX

Create interactive Power BI and Streamlit dashboards to show real-time carbon footprints by faculty and provider.

The screenshot shows a Trello board with two cards:

- E5-US2 | Interactive Filters by Date and Provider**: A card with a red "Must" label, due on 18 Dec, 22:39, and 5 story points. It has sections for Description, Acceptance Criteria (listing requirements like "Data and Provider filters work across all pages"), and Definition of Done (listing tasks like "Filter logic tested in Power BI Service").
- E5-US3 | PDF Export and Automated Reporting**: A card with a green "Should" label, due on 19 Dec, 22:39, and 5 story points. It has sections for Description (listing requirements like "PDF export maintains layout and data accuracy 100%"), Acceptance Criteria (listing requirements like "Export workflow triggered via Logic App weekly"), Amazing Fields (mentioning permission needs), and Definition of Done (listing tasks like "Export functionality tested for 5 sample reports").

E5-US1	As a BI Dev, I want a live dashboard so that users see emissions in real time.	Must	3	Refresh <3s.
E5-US2	As a User, I want filters by date/provider.	Must	3	All filters validated.
E5-US3	As a Sustainability Officer, I want PDF export for reports.	Should	4	100% fidelity.
E5-US4	As a Front-end Dev, I want to embed Power BI in the portal.	Could	4	Responsive frame.

Epic-6 Alerts & Automation

Implement Logic Apps and Azure Functions to notify threshold breaches and trigger remediation tickets.

○ E6-US1 | Email Alerts for CO₂ Threshold Breach

+ Add Checklist Members Attachment

Labels Due date Story Points
Must + 22 Dec, 22:39 5

Description Edit

As an IT Admin, I want automatic email alerts when CO₂ usage exceeds set thresholds so that issues can be addressed immediately.

Acceptance Criteria:

- Logic App triggers email within 2 minutes of breach detection.
- Threshold value configurable per department and provider.
- Email contains timestamp, metric value and source system.
- Alerts sent to Department Head and Sustainability Officer.
- Alert delivery success rate $\geq 99\%$.

Amazing Fields Visible Fields

Amazing Fields needs permission to access card data on this board. We use the minimum permissions needed to provide access and support. You can cancel authorization at any time.

For more details see: [Amazing Fields Permissions](#)

Authorize with Trello

DoD Delete

SMTP configuration validated in QA and Prod.
 Alert emails tested with sample breach data.
 Logs stored in monitoring table.
 Peer review and approval completed.

○ E6-US2 | Automatic Jira Ticket Creation via Logic App

+ Add Checklist Members Attachment

Labels Due date Story Points
Should + 22 Dec, 22:39 5

Description Edit

As an Automation Engineer, I want a Logic App that creates Jira tickets when critical alerts fire so that incidents are tracked formally.

Acceptance Criteria:

- Jira API integration established with auth token rotation.
- Tickets include alert ID, timestamp, severity and description.
- Ticket auto-assigned to Infra team based on severity.
- Confirmation email sent on ticket creation.
- Workflow success rate $\geq 98\%$.

Amazing Fields Visible Fields

Amazing Fields needs permission to access card data on this board. We use the minimum permissions needed to provide access and support. You can cancel authorization at any time.

For more details see: [Amazing Fields Permissions](#)

Authorize with Trello

DoD Delete

Logic App flow tested end-to-end.
 Ticket creation logs verified in Jira.
 QA approved automation documentation.

E6-US1	As an IT Admin, I want email alerts for CO ₂ breach.	Must	4	Alert <2min.
E6-US2	As an Automation Engineer, I want a Logic App to create Jira tickets.	Should	5	Ticket auto-logged.
E6-US3	As a QA Analyst, I want alert events logged.	Could	5	Audit table complete.
E6-US4	As a DevOps Engineer, I want a test-alert simulator so that rules verify pre-prod.	Could	5	Email + Jira <2min; logged.

Epic-7 Governance & Security

Ensure data privacy and compliance through Key Vault, RBAC, and Purview integration.

E7-US1 | RBAC Roles for Staff and Students

+ Add Checklist Members Attachment

Labels	Due date	Story Points	
Must	+	25 Dec, 22:39	7

Description Edit

As a Security Engineer, I want role-based access control (RBAC) so that data access is restricted according to user roles.

Acceptance Criteria:

- Azure RBAC policies configured for Department Head, Researcher, IT Infra Admin and Financial Head.
- Least-privilege principle applied to all roles.
- Access logs captured for every login and data query.
- Unauthorized access attempts alert Security group within 5 minutes.
- Access review performed monthly via Purview report.

Amazing Fields Visible Fields

Amazing Fields needs permission to access card data on this board. We use the minimum permissions needed to provide access and support. You can cancel authorization at any time.

For more details see: [Amazing Fields Permissions](#)

[Authorize with Trello](#)

DoD Delete

0% 0%

RBAC matrix approved by Security Lead.

Validation scripts executed and logged in Confluence.

Peer review completed and merged to main.

Product Backlog ▾

E7-US3 | Data Lineage Tracking in Azure Purview

+ Add Checklist Members Attachment

Labels	Due date	Story Points	
Should	+	26 Dec, 22:39	7

Description Edit

As a Compliance Officer, I want end-to-end data lineage so that the movement of CO₂ data is fully traceable.

Acceptance Criteria:

- Purview scan configured for ADLS, ADF, Synapse and Power BI.
- Lineage diagram generated for Bronze→Silver→Gold flow.
- Sensitive fields tagged as "Confidential."
- Monthly lineage report auto-exported to SharePoint.
- Access restricted to Compliance and Security teams.

Amazing Fields Visible Fields

Amazing Fields needs permission to access card data on this board. We use the minimum permissions needed to provide access and support. You can cancel authorization at any time.

For more details see: [Amazing Fields Permissions](#)

[Authorize with Trello](#)

DoD Delete

0% 0%

Purview dashboard reviewed by Compliance Officer.

QA verified tags and metadata accuracy.

Scan schedule validated and logged.

E7-US1	As a Security Eng, I want RBAC roles for staff/students.	Must	2	Access matrix validated.
E7-US2	As a Cloud Admin, I want secrets stored in Key Vault.	Must	1	No plain-text keys.
E7-US3	As a Compliance Officer, I want data lineage tracked in Purview.	Should	5	Lineage diagram.
E7-US4	As a Compliance Officer, I want automated data-retention policies so that old CO ₂ data is purged securely after 12 months.	Should	6	Monthly job; deletion logged.

Epic-8 Performance & Sustainability Monitoring

Use App Insights and Cost Management API to monitor uptime, efficiency, and carbon savings against targets.

E8-US1 | Runtime Metrics via Application Insights

+ Add Checklist Members Attachment

Labels Due date Story Points
Must + 1 Jan 2026, 22:39 ✓ 7

Description As a DevOps Engineer, I want to collect runtime metrics through Azure Application Insights so that I can monitor system health and performance.

Acceptance Criteria:

- Application Insights integrated across ADF, Synapse, and Logic Apps.
- Metrics captured: CPU, memory, request rate, failure count, and latency.
- Custom dashboard built for real-time telemetry.
- Alert threshold set for latency > 3 s or error rate > 1 %.
- Uptime target ≥ 99.5 % with daily health reports.

Amazing Fields Visible Fields

Amazing Fields needs permission to access card data on this board. We use the minimum permissions needed to provide access and support. You can cancel authorization at any time.

For more details see: [Amazing Fields Permissions](#)

Authorize with Trello

DoD

Integration validated in staging and production.
Dashboard verified by IT Infra Admin.
Logs reviewed weekly by DevOps team.
QA sign-off recorded in Jira.

E8-US2 | Weekly Cost and Usage Reports

+ Add Checklist Members Attachment

Labels Due date Story Points
Should + 2 Jan 2026, 22:39 ✓ 5

Description As a Financial Analyst, I want automated weekly cost and usage reports so that cloud spend can be tracked against emission metrics.

Acceptance Criteria:

- Azure Cost Management API integrated into data pipeline.
- Report generated every Monday 9 AM and emailed to stakeholders.
- Report includes cost by provider, service, and department.
- Variance ≥ 10 % week-to-week highlighted automatically.
- Stored securely in SharePoint folder.

Amazing Fields Visible Fields

Amazing Fields needs permission to access card data on this board. We use the minimum permissions needed to provide access and support. You can cancel authorization at any time.

For more details see: [Amazing Fields Permissions](#)

Authorize with Trello

DoD

Report delivery validated for three consecutive weeks.
Financial Head approved layout and accuracy.
Logic App scheduler tested and documented.

E8-US1	As a DevOps Eng, I want runtime metrics from App Insights.	Must	4	Uptime ≥99.5%.
E8-US2	As a Finance Analyst, I want weekly cost reports.	Should	4	Auto-email Monday.
E8-US3	As a Sustainability Officer, I want carbon reduction KPIs displayed.	Could	6	≥3 KPIs shown.
E8-US4	As an Ops Lead, I want the response time benchmarked.	Could	6	<3s average.

3.2. Sprint Allocation

Team Capacity

Team	Working	Hours per	Total	Other Factors	Story Point

Member	Days	Day	Hours		Capacity
Cong	10	6	60	PO Overhead: Reduced coding hours for meetings and backlog.	3-4
Bhavesh	10	8	80	High Proficiency: Specialized experience boosts velocity	5-6
James	10	8	80	Standard developer	4-5
Ma	10	8	80	Standard developer	4-5
Total Sprint Capacity			300		16-20

Sprint Planning

Sprint	Sprint Goal	Assigned User Stories (ID & SP)	Total SP	Primary Focus & Pain Point Resolution (Justification)	Contingency & Adaptability Notes (Potential Issues)
Sprint 1	Securely establish the foundational, authenticated multi-cloud data ingestion pipeline.	E1-US1 (5), E1-US2 (4), E7-US2 (3), E2-US2 (4), E1-US3 (2)	18	Secure Foundation. Establishes secure access (E7-US2) and Ingestion (E1-US1). E2-US2 ensures data efficiency. Dependency: Crucial prerequisite for all subsequent work.	E1-US3 (Streaming) is the buffer; drop to accept hourly refresh if secure access is blocked.
Sprint 2	Build the validated Star Schema Data Warehouse to ensure data quality and trust for downstream analytics.	E2-US1 (4), E2-US3 (2), E3-US1 (5), E3-US2 (5), E7-US1 (3)	19	Data Integrity & Warehouse. Focuses on data quality (E2-US1) and building the central analytical backbone (E3-US1/US2). E7-US1 (RBAC) is prerequisite for dashboard in Sprint 3.	E2-US3 (Schema Logs) is buffer. Must protect the Must-Have core DW structure (E3-US1/US2) for reporting.

Sprint 3 <u>(Release 1)</u>	Deliver the MVP live dashboard and commence the carbon emission forecasting model.	E3-US3 (3), E3-US4 (2), E5-US1 (5), E5-US2 (4), E4-US1 (5)	19	MVP Dashboard. Delivers Prof. Boehly's Live Dashboard (E5-US1/US2). Starts E4-US1 for Sarah (Finance).	Defer E3-US3/E3-US4 to ensure the Live Dashboard MVP is fully delivered if E4-US1 (Forecast Model) proves complex.
Sprint 4	Operationalize the system by implementing key alerts, optimization recommendations, and performance monitoring.	E4-US2 (4), E5-US3 (3), E6-US1 (4), E8-US1 (4), E8-US2 (3)	18	Actionable Insights & Ops. Focuses on features for John Terry (E6-US1, E4-US2) and Finance reports (E8-US2).	Release Cut Point: E4-US2 and E6-US1 are deferred to Release 2 if scope is cut after Sprint 3, preserving core reporting.

3.3. Release Phase 1

Element	Description	Value Delivered
Epics Included	Epic 1 (Ingestion), Epic 7 (Security), Epic 2 (Cleansing), Epic 3 (Warehouse), Epic 5 (Visualization), and Epic 4 (Forecasting Start).	A fully operational pipeline, from data acquisition to visualization and initial predictive modeling.
Delivery Point	End of Sprint 3	Delivers the core system needed to derive and view the first key business metric (Carbon KPI).
Must-Have Stories	E1-US1/US2, E7-US1/US2, E2-US1, E3-US1/US2, E5-US1/US2, E4-US1.	These stories are non-negotiable for achieving a secure, verified, and visible Carbon KPI.

3.4. Sprint 1 Execution Plan

Sprint Goal

Establish the foundational, authenticated multi-cloud data ingestion pipeline securely, covering **Epic 1 (Data Ingestion)**, **Epic 7 (Security)**, and **Epic 2 (Data Transformation)**.

Justification

Builds the secure data foundation necessary to provide trusted KPIs for Prof. Boehly and non-negotiable security controls for John Terry, validating the project's core multi-cloud vision.

ID	User Story	MoSCoW	Estimated SP	Acceptance Criteria
E1-US1	As a Data Engineer, I want ADF to ingest AWS/Azure/GCP CO ₂ so that data is centralised in ADLS.	Must	5	APIs authenticated; hourly load < 5 min latency; log summary generated.
E1-US2	As a Cloud Architect, I want API Management with token rotation so that access is secure and auditable.	Must	4	Token \leq 7 days; audit log complete; alerts < 5 min.
E7-US2	As a Cloud Admin, I want secrets stored in Key Vault.	Must	3	No plain-text keys.
E2-US2	As a Data Engineer, I want JSON → Parquet so that queries run faster.	Must	4	Query speed \geq 3x improvement.
E1-US3 (Buffer)	As a Data Engineer, I want Event Hub streaming so that dashboards update quickly.	Should	2	95% events within 2 min; duplicates < 1%.
Total Story Points			18 SP	

Contingency and Buffer

- **Capacity Buffer:** The team has a capacity of 20 SP. 2 SP buffer is available
- **Contingency:** If technical risks, such as unexpected API throttling or complex authentication, block the Must-Have security (E7-US2) or ingestion (E1-US1), E1-US3

will be postponed. Protecting the core deliverables by prioritizing security and data availability over the non-essential features.

Definition of Done (DoD)

A user story is considered complete when all acceptance criteria and DoD are met and verified.

- **Testing:**
 - Peer-reviewed and unit-tested with >80% code coverage.
- **Integration:**
 - Feature is successfully integrated into the staging environment.
 - Data pipelines process correctly through the Bronze layer with validated schemas.
- **Documentation & Approval:**
 - Documentations are finalised.
 - Demonstrated and approved by the Product Owner.

4. Reflection

Framework & Scope

As students, we reflect on our experience in the Agile project, utilizing Gibbs' reflective cycle to examine our learning journey.

Description — What We Experienced as Students

As students new to Agile methodologies, our first step was learning to use Trello as a Scrum board to visualize our workflow across two-week sprints. We practiced MoSCoW prioritization and defined shared "Done" criteria for user stories—all novel concepts introduced in this course. Daily stand-ups and sprint retrospectives became our practical labs for understanding Agile rhythms. The Week 11 interview and Week 12 demonstration served as key learning milestones where we received direct feedback on our application of these concepts.

Evaluation — Our Learning Successes and Struggles

What enhanced our learning: Using Trello to visualize our simulated product backlog transformed abstract Agile concepts into something concrete. Moving user story cards from "To Do" to "Done" helped us understand sprint flow in a way reading the Scrum Guide didn't as we could literally see how work progresses through stages. Planning Poker for estimating our CO₂ emission measurement features revealed how different our individual assumptions were. Defining our Definition of Done gave us practical quality standards rather than vague "completion," which improved our report deliverable consistency.

Where we struggled as learners: Early on, we found ourselves reverting to familiar predictive planning habits, debating future designs instead of focusing on delivering working increments. We also learned the hard way about dependencies and capacity planning when our first sprint experienced task rollover, giving us firsthand experience of a common project management challenge. One challenge emerged during the assessment interview when we didn't expect some scenarios, realizing that translating Agile principles to spontaneous problem-solving situations was more difficult than anticipated.

However, our demonstration component performed well because we'd thoroughly prepared core Agile definitions, showing structured preparation helped us communicate our learning effectively.

Analysis — Connecting Experience to Course Concepts

Our progression followed a clear learning curve that aligns with Tuckman's model of group development:

- During Forming, we elected a leader who facilitated task allocation, with members volunteering for preferred roles—this initial structure felt efficient and collaborative.
- However, Storming emerged when debating our report outline. A 30-minute argument revealed unclear task priorities, causing all members to draft overlapping sections. For example, one member thought to put specific KPI into Project Vision while another member disagreed as this section should be kept generic. This wasted approximately 2 hours of collective effort and highlighted how theoretical understanding doesn't automatically translate to coordinated execution.

- Transitioning to Norming required pausing to explicitly discuss priorities and outline structure together. Implementing a "one writer, one reviewer" system eliminated duplication while ensuring quality control.
- By Performing, the transformation was evident: we no longer hesitated to raise concerns or suggest improvements, trusting our voices would be heard rather than dismissed. This psychological safety enabled smooth collaboration and consistent report quality that met assessment criteria. Critically, this progression taught me that high-performing teams aren't conflict-free but they're built through intentionally working through earlier disagreements to establish mutual trust.

Key learning gaps we identified included

- Balancing discovery discussions with delivery focus—a challenge noted in the PMI Agile Practice Guide (2017) as common in early Agile adoption
- Understanding the importance of preparation criteria before starting work
- Learning to plan for uncertainties while maintaining progress

Conclusion — What We Learned

- Adaptive thinking over rigid planning: We learned that maintaining transparency through tools like Trello and embracing change produces more realistic outcomes than extensive upfront planning.
- The value of incremental progress: Delivering thin vertical slices taught us how to identify and address integration issues early—a crucial skill for future projects.
- The importance of structured reflection: Using frameworks like Gibbs' cycle transformed our experience from simple description to meaningful learning, helping us extract maximum value from both successes and challenges.
- These outcomes reflect the emphasis the Scrum Guide places on iterative progress and adaptation as foundations for effective teamwork (Schwaber & Sutherland, 2020)—principles we now understand through direct experience.

Action Plan — Applying Our Learning Beyond This Course

- Timebox meetings with accountability: Set strict 15-minute limits for standups with a rotating timekeeper role; use a parking lot document for off-topic discussions requiring deeper exploration later.
- Create task-ready checklists: Before claiming any task, verify it has clear acceptance criteria, necessary resources identified, and no unresolved dependencies—applying Definition of Ready systematically.
- Build realistic capacity buffers: Commit to only 70% of estimated capacity in sprint planning, explicitly reserving 30% for coursework conflicts, unexpected complexity, and revision time.
- Track and review velocity data: Record story points completed vs. committed each sprint; review trends before planning sessions to calibrate estimates based on actual performance patterns.
- Rehearse under pressure: Conduct timed mock interviews with unexpected questions from peers outside the team to practice articulating concepts spontaneously rather than from scripts.

- Rotate facilitation weekly: Assign different members to lead planning, standups, and retrospectives each sprint, ensuring everyone develops both leadership and followership skills through practice.
- Map dependencies during planning: Visually diagram task relationships and identify blockers before sprint commitment; assign a "blocker monitor" to flag issues in daily standups proactively.
- Define "demo-ready" incrementally: Ensure every sprint produces something presentable to stakeholders, even if incomplete, forcing us to prioritize integrated, testable work over fragmented tasks.

5. References

- AXELOS. (2019). *ITIL Foundation: ITIL 4 edition*. TSO (The Stationery Office).
- ISO/IEC 25010:2011. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. International Organization for Standardization. <https://www.iso.org/standard/35733.html>
- ISO 14064-1:2018. *Greenhouse gases — Part 1: Specification with guidance at the organization level for quantification and reporting of greenhouse gas emissions and removals*. ISO. <https://www.iso.org/standard/66453.html>
- Microsoft Azure Well-Architected Framework. (2023). *Reliability and operational excellence pillars*. <https://learn.microsoft.com/en-us/azure/architecture/framework/>
- Monash University. (2021). *Monash University strategic plan 2021-2030: Impact 2030*. https://www.monash.edu/_data/assets/pdf_file/0009/2680389/Strategic-Plan_Impact-2030_Sept-2025.pdf
- Nielsen, J. (2010). *Website response times*. Nielsen Norman Group. <https://www.nngroup.com/articles/website-response-times/>
- Project Management Institute. 2017. *Agile Practice Guide*. <https://www.pmi.org/standards/agile-practice-guide>
- Schwaber & Sutherland. 2020. *The Scrum Guide™*. <https://scrumguides.org/>
- Think with Google. (2016). *The need for mobile speed*. https://www.thinkwithgoogle.com/_qs/documents/2340/bc22e_The_Need_for_Mobile_Speed_-_FINAL_1.pdf

6. GenAI Declaration

We confirm that this is our original work. We have used ChatGPT solely to help improve the clarity and expression of our writing.