DEVELOPMENT OF AN IMAGE-BASED PLANT DISEASE DIAGNOSIS APPLICATION VIA JIRA

MSPM- FALL 2025

03-398242-007 MUHAMMAD MANSOOR

03-398242-016 MOHAMMAD ALI MIR

A project submitted in fulfilment of the
requirements for the completion of the course Advance Skills for Project Management Professionals

Course Instructor: Muhammad Mohsin Rashid

Department of Management Sciences

Bahria University Lahore Campus

December 2025

APPROVAL FOR EXAMINATION

Student Name:

Registration No.:

Student Name:

Registration No.:

Programme of Study: MSPM

Project Title:

It is to certify that the above student's project has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for examination. I have also conducted a plagiarism test of this thesis using HEC prescribed software and found similarity index _____% that is within the permissible limit set by the HEC for the MS degree thesis. I have also found the project in a format recognized by the BU.

Course Instructor Signature: _____

Date: _____

Name: _____

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-398242-007	MUHAMMAD MANSOOR	
03-398242-016	MOHAMMAD ALI MIR	

Date: Dec 31, 2025

DEDICATION

Specially dedicated to

our parents and teachers

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our project Instructor, Sir Mohsin Rashid for his invaluable advice, guidance, and his enormous patience throughout the project.

In addition, we would also like to express our gratitude to our loving parents and friends who had helped and given us encouragement.

MUHAMMAD MANSOOR

MOHAMMAD ALI MIR

EXECUTIVE SUMMARY

The Image-Based Plant Disease Diagnosis Application was a project commissioned by the Marriam Rural Welfare Organization (MRWO) to deal with the shortage of agricultural expertise in the Bahawalpur and Lodhran. The project uses Artificial Intelligence to enable farmers to identify Cotton and Wheat diseases instantly and, thus, directly mitigate losses of crops and reliance on broad-spectrum pesticides.

The sprints were conducted during the period of 1 to 29 December, 2025, via Jira. The scrum team operated at a steady velocity of 40 Story Points per sprint, and 50 Points in bug-intensive sprints. It was this adaptability in planning that helped the team to achieve tight deadlines and deal with complicated delivery problems. The solution is based on industry standards of mobile and cloud technology and is optimized to operate in remote and low-connectivity environments.

The project was able to provide a very accurate (>92%) diagnostic mobile application and Disease Trends Dashboard to the MRWO leadership. The four Sprints were completed within the planned timeframe. Major technical risks such as hardware limitations and latency of the system were addressed in advance, without affecting the schedule. On the financial side, the project was closed within the PKR 1,347,500 budget. This endeavor demonstrated the value of agile methodologies in social projects.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: PROJECT INTRODUCTION

1.1 Organizational Overview: Marriam Rural Welfare Organization (MRWO)

This section focuses on Marriam Rural Welfare Organization (MRWO).

1.1.1 Vision

The vision of Marriam Rural Welfare Organization (MRWO) is to have a just, equal and humane society where no individual will be deprived of their basic human rights regardless of his or her socio-economic status. This includes access to quality education, affordable as well as reliable health care, clean water, sustainable livelihoods as well as safe and ecologically acceptable environment. The MRWO has a vision of a world in which poverty, social exclusion, and environmental degradation are a thing of the past and development is inclusive and sustainable. The company believes in being a beacon of hope to the underprivileged and bringing change to individuals and communities through collective empowerment and sustainability.

1.1.2 Mission

MRWO's mission is to establish an ethical and empowering culture by providing a voice to marginalized groups, especially women, children, and the youth. To bridge the gap between policy and practice through strategic partnerships with governments, civil societies, international development actors and other local mobilizers. And involving the underprivileged communities in all stages of developmental activities.

1.1.3 Core Values

- **Inclusivity:** MRWO embraces the idea of making development inclusive and a right of all. The organization is dynamic and active on both ends of the spectrum, regardless of the individual or group's caste, creed, ethnicity, religion, gender, age, and disability.
- **Transparency:** MRWO is founded on transparency. The organization, over the years, has been transparent to stakeholders in all levels of financial management and project implementation.
- **Empowerment:** Instead of only providing aids, MRWO empowers people and communities to be actors of their own change.
- **Integrity:** MRWO is committed to the highest standards of ethics, moral responsibility and honesty.
- **Sustainability:** MRWO recognizes the fact that people and environment are related; hence, sustainability is embraced in all its operations.
- **Teamwork:** MRWO values the power of teamwork. It believes that only the organic collaboration of communities, government agencies, non-profit organizations, academia, and international development partners will allow bringing change in the long-term.

1.1.4 Financial Details & Company Strengths

Financial Capacity

MRWO has a robust financial foundation that supports smooth running of its multi-dimensional development programs. The annual turnover of the organization ranges between PKR 3 to 5 million through the help of donor funds, grants and project-related funds. However, MRWO has been able to repeatedly deliver projects worth over PKR 20.5 million, which shows its scalability, financial discipline and ability to handle large scale projects.

Company Strengths

- **Experience in Development Sector:** MRWO has over 20 years of experience in the development sector, resolving the social, economic, and cultural problems of the rural population.
- **Human Resources:** The organization has a small team of 18 highly skilled professionals. In addition, MRWO workforce includes over 230 volunteers.
- **International Collaboration:** MRWO has had an opportunity to collaborate with well-known organizations like WWF (environmental conservation) and UNICEF (childcare).
- **Robust Monitoring, Evaluation and Governance Framework:** The company has strong monitoring and evaluation (M&E) processes that ensure transparency and impact evaluation.
- **Deep Grassroot Penetration:** MRWO has developed a strong bond with the local communities especially in underserved Bahawalpur and Lodhran.

1.1.5 Organization Portfolio

MRWO is currently active in:

- Education & Literacy
- Health & Family Planning
- Women Empowerment
- **Sustainable and Regenerative Agriculture** (This project belongs to this domain).
- Environmental Protection
- Water, Sanitation and Hygiene (WASH).
- Skill Generation and Revenue.
- Emergency Relief and Rehabilitation.
- Child Protection and Eradication of Labor.

- Democratic Awareness and Voter Education.

1.2 Project Introduction & Need

Although there has been considerable progress in agricultural intervention and methods, the rural areas of Pakistan are long awaiting technological integration. Particularly, diseased crops in widespread fields exert an undue burden on farmers, field workers, and agricultural extension departments. We live in an age of rapid technological innovation and development. There is an increase in adoption of Digital Agriculture (AgriTech), which has been proven to improve the quality of both yield and food security.

Plant disease diagnosis is a necessity for any farmer who wishes agricultural success. Nevertheless, the process of diagnosing plant diseases by visual examination of the field is a lengthy and trying task that requires guesswork, delayed consultations with experts, and the expensive purchase of broad-spectrum pesticides. Thus, there is a gap and need for a convenient and user-friendly service that facilitates easy and accessible diagnosis of crop diseases.

In this project, we focused on creating an application called MRWO Plant Doctor, which can be used to diagnose plant diseases. The application enables users to take a picture of the plant leaf which is compared against a database of diseases for Cotton and Wheat. Once the picture is analyzed, the diagnosis, confidence score, and recommended treatments are immediately available to the users. The app we created will allow MRWO managers to regulate data, see trends of diseases on a dashboard, and organize relief operations in case a large outbreak is observed.

1.3 Problem Statement

In Bahawalpur and Lodhran, farmers have been incurring heavy economic losses because of late detection of diseases such as Cotton Leaf Curl Virus (CLCV) and Wheat Rust. To make matters worse, the ratio of agricultural experts to farmers is very low. By the time an expert is able to visit a field, the disease has spread, killing the harvest. Through this

project, will put the “Expert” in the pocket of every farmer with the help of inference from Artificial Intelligence (AI).

CHAPTER 2: SCRUM

Based on the 2025 Scrum Body of Knowledge, the Scrum framework (we used) consists of the following 5 phases: Initiate, Plan and Estimate, Implement, Review and Retrospect and Release.

2.1 Initiate

2.1.1 Create Project Vision

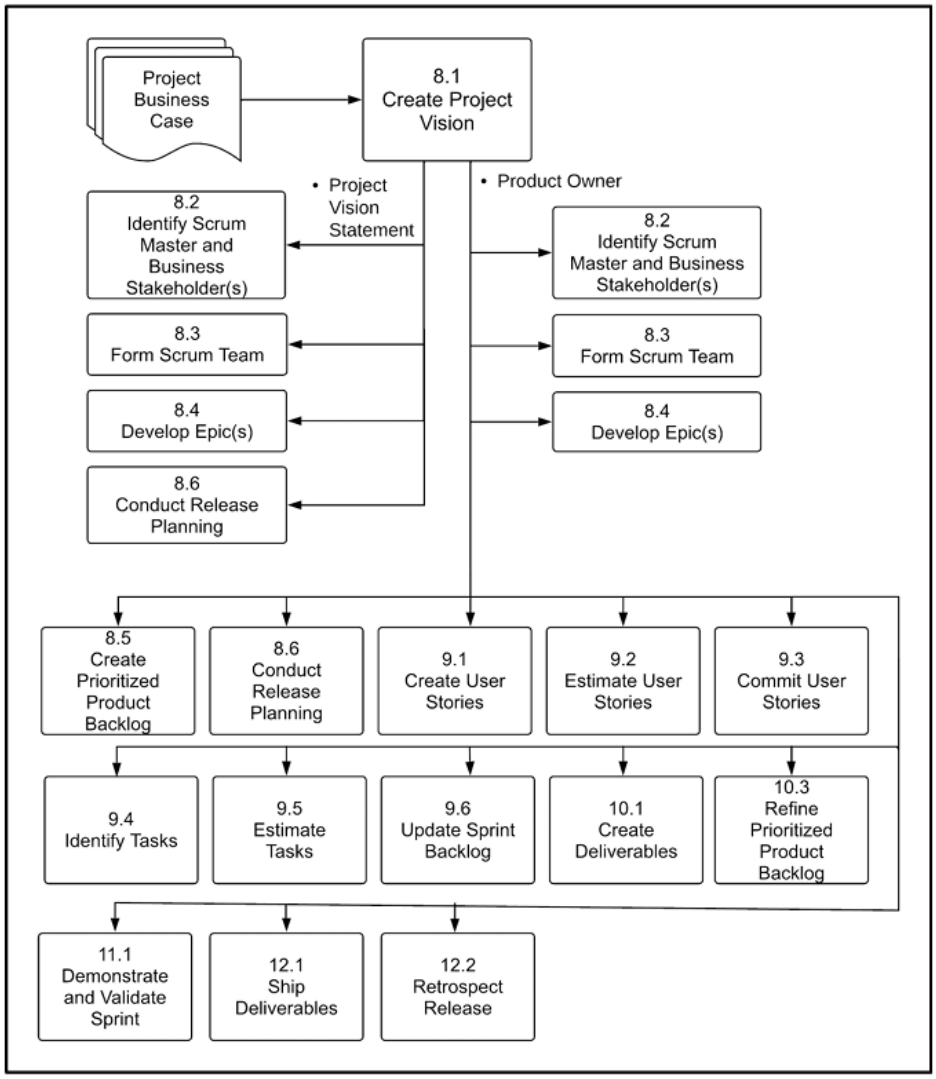


Figure 1. Data Flow Diagram for Create Vision Process (Satpathy, 2025)

The Scrum project began with the process of creating the project vision. During this process, the project business case was used to prepare a Project Vision Statement that was a source of inspiration and focus for the whole project. The person responsible for this process was the Product Owner (Mr. Ali Mir.)

2.1.1.1 Product Owner

The Product Owner is the individual responsible of maximizing value of the product that is generated due to Scrum. His duties include keeping the stakeholders involved in the project and communicating the product vision, objective and backlog to the Scrum team. The product owner is also responsible for managing and prioritizing the backlog. The following are basic details about the Product Owner for our project:

- Name: Mr. Ali Mir
- Designation: Product Owner
- Contact No: [Confidential]
- Email: ali.mir@email.com

2.1.1.2 Project Vision Statement

The vision statement for this project was as follows:

“To bring digital expertise in the hands of the everyday farmer by eliminating the delay and uncertainty in crop disease diagnosis through creation of an easy-to-use and accurate, AI-based App. This will not only assist in minimizing losses in crops among the rural communities but also enhance sustainable farming practices through by minimizing use of pesticides.”

2.1.1.3 Project Mission

Derived from the Project Vision, the Project Mission was as follows:

“MRWO Plant Doctor is an android-based application, which aims is to connect farmers to an effective diagnostic application. It enables the user to take pictures of crop leaves

with ease and gets instant identification of diseases like the Rust or Leaf Spot. The App also allows users to make learned decisions about their crops giving them access to treatment protocols. MRWO Plant Doctor is focused on enhancing the yield and economic stability of Pakistani farmers by enabling users to be in control of the health of their own fields."

2.1.1.4 Project Charter

The project Charter for this project was as follows:

Table 1. Project Charter

Project Title	An Image-based Plant Disease Diagnosis Application
Project Start Date	December 1, 2025
Project Finish Date	December 29, 2025
Project Budget	PKR 1,347,500
Project Objective	Create a user-friendly image-based plant disease diagnosis application for farming community in Pakistan. s
Key Deliverables	**Validated Image Dataset:** Approximately 3,000 expert-verified images of Wheat (Healthy, Rust, Leaf Spot) and Cotton (Healthy, CLCV, Bacterial Blight). **Trained AI Model:** CNN model that has more than 90% validation accuracy. **Backend API System:** Secure cloud infrastructure linking app and AI model. **Mobile Application:** Android User Interface for capture and diagnosis. **User Manual:** Usage and troubleshooting guide for farm managers and farmers.
Project Team Structure	**Product Owner:** Mr. Ali Mir (Defines vision, prioritizes backlog). **Scrum Master:** Mr. Muhammad Mansoor (Facilitates sprints). **Dev Team:** Agri. Researcher, Software Engineer, Backend Developer, Farm Manager (Execute technical implementation).

Success Criteria	Completion on time and within budget, meeting MRWO requirements. AI model must achieve greater than **90% validation accuracy**.
High Level Risks	**GPU Memory:** Training size might exceed local hardware capabilities; cloud scaling is required. **Connectivity Issues:** Poor internet connectivity in rural areas; Requires offline Mode and image compression. **Diagnostic Inaccuracy:** Validate images on ground truth data and provide user disclaimer. **Latency:** Delays caused by high traffic; Requires hot-loading and caching.
Project Methodology	Product Owner gathers the requirements from Sponsor (MRWO) and briefs the team. Design and testing are performed by the cross-functional development team. Product Owner works with Sponsor to align deliverables (increments) with acceptance criteria. Scrum ceremonies to be held according to the calendar plan. All the activities in the project are to be monitored by Product Owner and Scrum Master.

2.1.1.5 Project Budget

We budgeted the developmental process based on factors like scope of work, platform compatibility, and the training requirements for the AI model. The app was planned and budgeted based on the following features:

- Image Capture and Upload
- AI Inference Engine (CNN)
- Disease Treatment Database
- Admin Dashboard for Trends

Based on these aspects, an approximate budget of PKR 1.3 million was allocated to this project. (See Annexure A for Cost Baseline).

2.1.2 Identify Scrum Master and Business Stakeholders

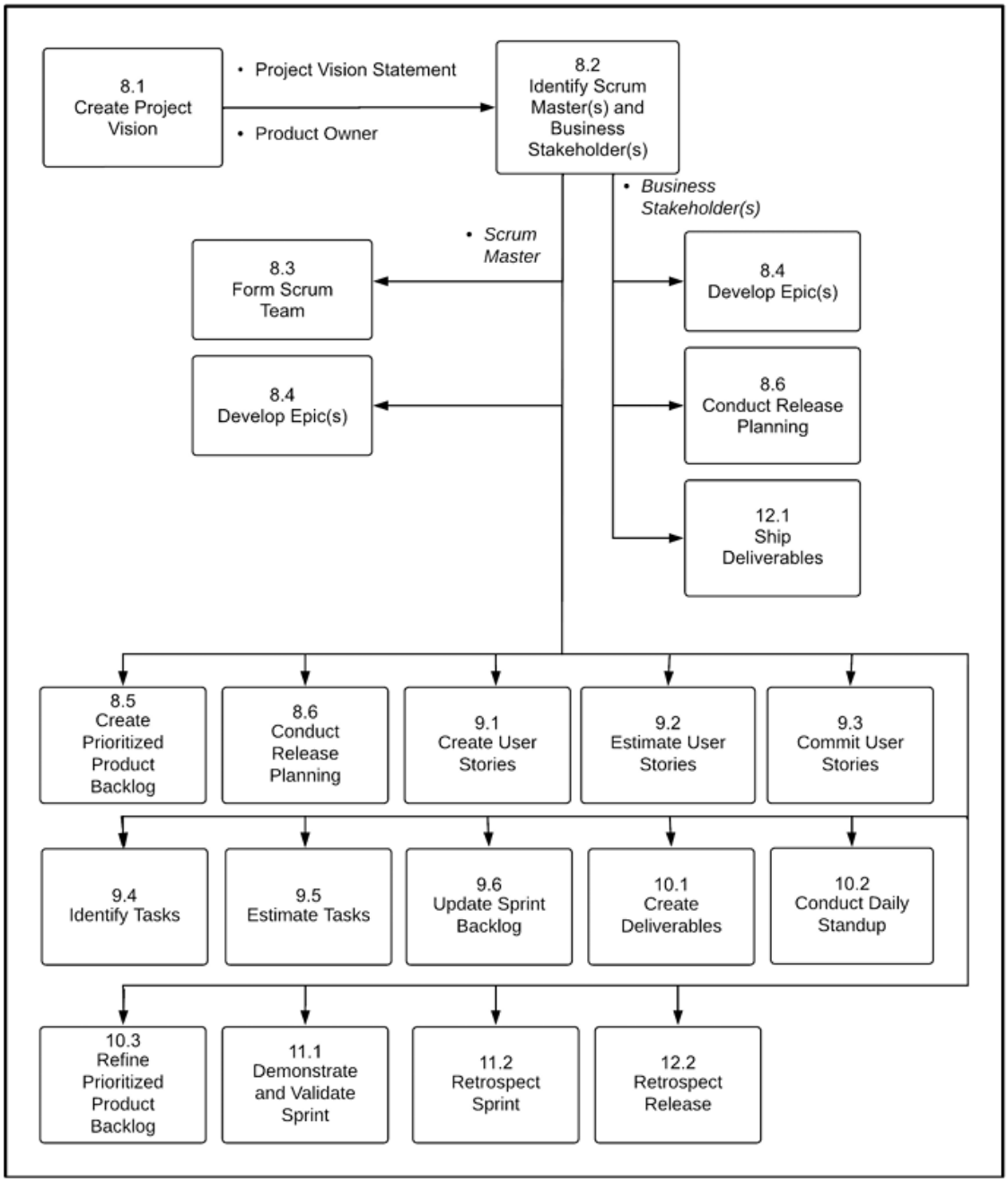


Figure 2. Data Flow Diagram for Identify Scrum Master and Business Stakeholder(s) Process (Satpathy, 2025)

2.1.2.1 Scrum Master

Scrum Master is the person who facilitates the implementation of Scrum in a project and assists the development team in knowing and following Scrum rules. He plays a pivotal role in the entire Scrum process by training the team members, eliminating barriers and

encouraging cooperation and openness. Following are the basic details of Scrum Master for this project:

- Name: Mr. Muhammad Mansoor
- Designation: Scrum Master
- Contact No: [Confidential]
- Email: m.mansoor@email.com

2.1.2.2 Business Stakeholders

Stakeholders are individuals with interests in the project, or those who wish to gain value out of the product. They also engage with the Scrum team on a regular basis and affect the Project during the product development lifecycle. Such individuals can be internal or external to the product creating organization. The following table describes various business stakeholders for our project.

Table 2. Business Stakeholders for Plant Disease App Project

Name	**Position**	**Internal/External**	**Project Role**	**Power**	**Influence**
MRWO Director	Director	Internal	Sponsor	High	High
WWF Rep	Partner	External	Observer	Medium	Medium
Local Farmers	Community	External	End User	Low	High

2.1.3 Form Scrum Team

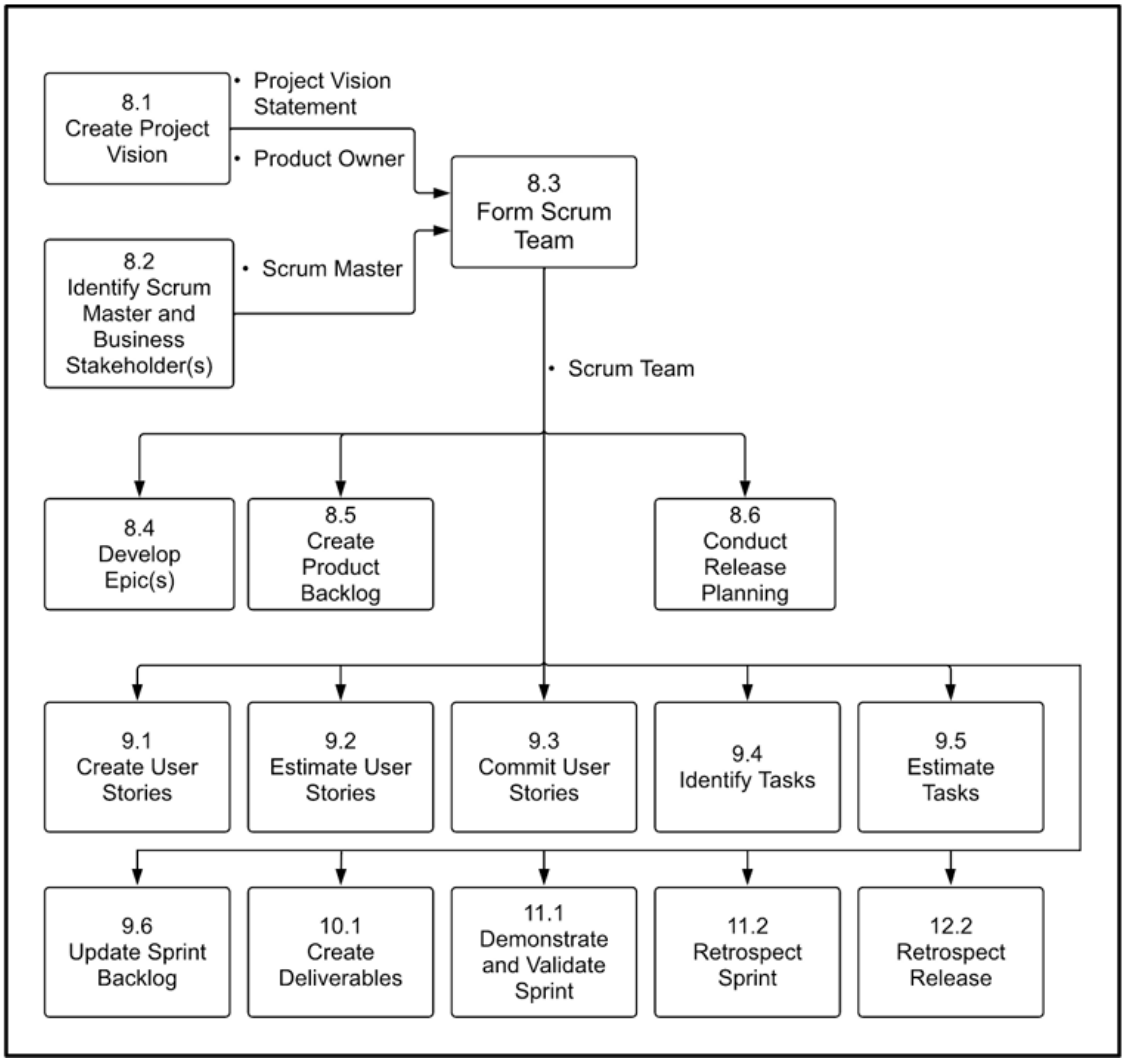


Figure 3. Data Flow Diagram for Form Scrum Team Process (Satpathy, 2025)

2.1.3.1 Scrum Team

The Scrum Team is also called the Development team. It is a group of individuals that collaborate for the design of a product via Scrum Framework. These are cross-functional and self-organizing teams. This team determines what features to include in each Sprint and approximate the effort to be put into the completion of each user story in that Sprint. In short, they are responsible for developing and delivering increments (deliverables).

Table 3. Scrum Team for Plant Disease App Project

Sr. No.	Name	Designation	Contact No.

01	Agricultural Researcher	Domain Expert	xxxx-3456789
02	Software Engineer	AI Developer	xxxx-7654321
03	Backend Developer	API Developer	xxxx-9876543
04	Farm Manager	UAT Tester	xxxx-0987654

2.1.3.2 Backups

Ideally, Scrum teams should have backups for critical developer roles. Such backups must preferably be from the same team to compensate for any unforeseen circumstances. The presence of backups for different tasks is crucial because it guarantees that there will be no major delay in productivity because of the absence of a key team member. With this in mind, the Software Engineer served as the backup for the Backend Developer and the Scrum Master would perform QA duties in case the Farm Manager was not available.

2.1.4 Develop Epics

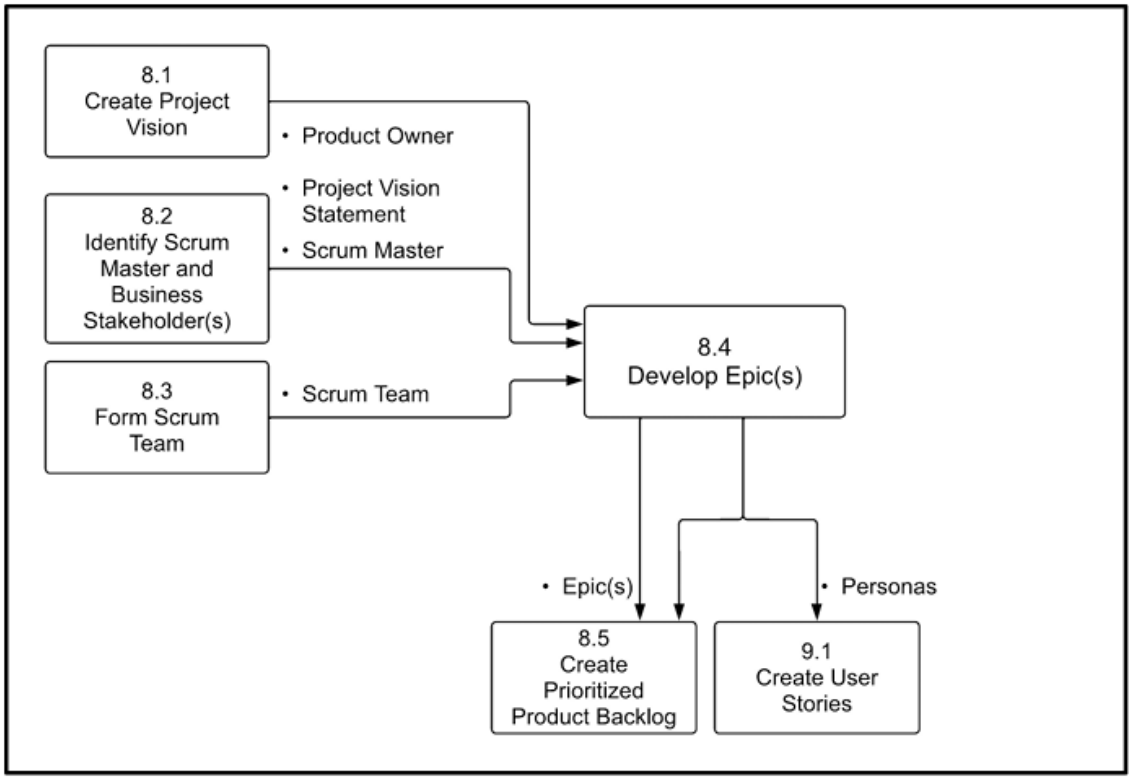


Figure 4. Data Flow Diagram for Develop Epics Process (Satpathy, 2025)

2.1.4.1 Epics

Scrum epics are large units of work. In this project, we determined two key Epics:

1. **MA-1: Model Development and Image Processing.**
    - Focus: The "Brain" of the system.
    - Included: Image Collection, Labeling, Preprocessing, CNN Model Training.
2. **MA-16: Development and Integration of Applications.**
    - Focus: The "Body" of the system.
    - Included: API Development, Mobile App UI, Dashboard, Deployment.

2.1.5 Create Prioritized Backlog

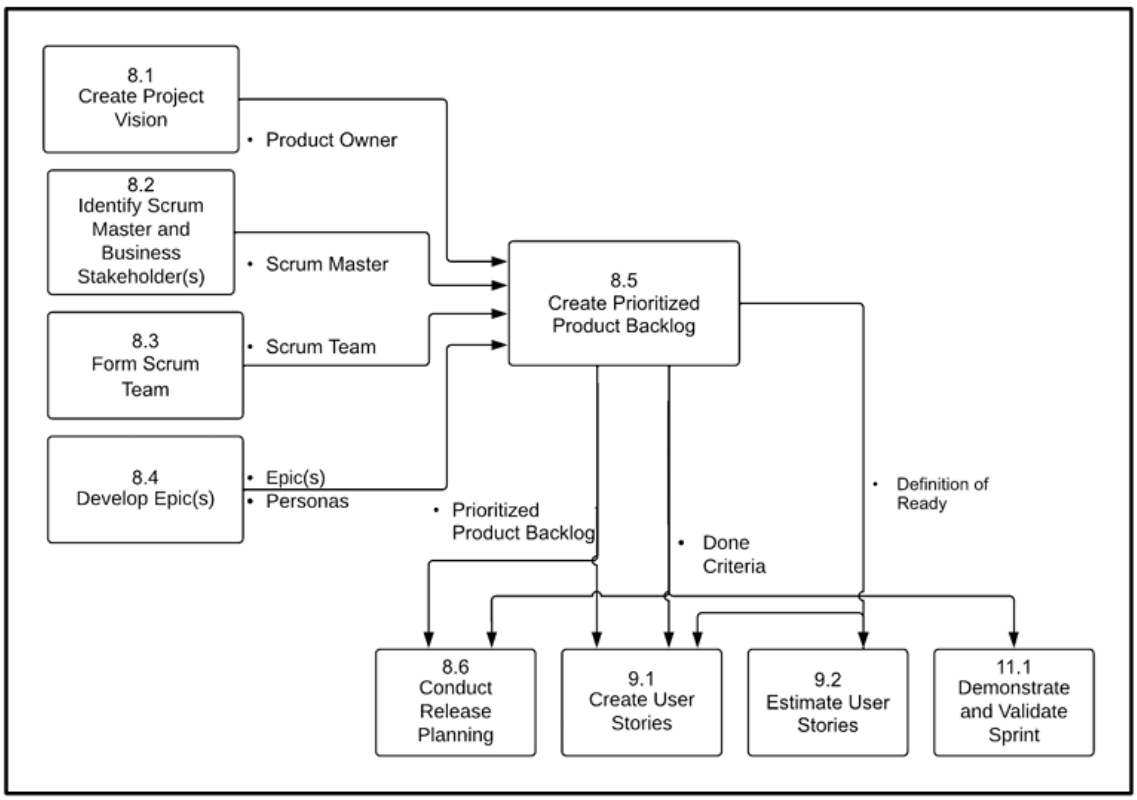


Figure 5. Data Flow Diagram for Create Prioritized Product Backlog Process (Satpathy, 2025)

2.1.5.1 Prioritized Product Backlog

Prioritized product backlog is a list of features and requirements needed to be a part of the product. They are ordered in terms of importance or value. These features or requirements are based on the product vision and the expectations of the customer and stakeholder (MRWO). The prioritized product backlog of the MRWO Plant Doctor App was as follows:

Table 4. Prioritized Product Backlog

Priority	**Feature Name**	**Priority**
1	Image Data Acquisition	Must Have
2	Dataset Annotation	Must Have
3	Image Preprocessing Pipeline	Must Have

4	AI Diagnostic Model	Must Have
5	Diagnostic API Service	Must Have
6	Performance Optimization	Must Have
7	Mobile Application Interface	Must Have
8	Disease Trends Dashboard	Should Have

2.1.5.2 Done Criteria

The done criteria for the MRWO Plant Doctor app are a list of requirements that an item in the product backlog or a product increment must satisfy to be considered done/completed. The definition of done helps the product owner and development team be sure that the work they produce is according to the expectations of customers and stakeholders. The criteria used in this project for the Definition of Done (DoD) was as follows:

1. The product increment is acceptable to the product owner.
2. The product increment is also tested adequately and appropriately (Unit Tests passed).
3. The Agricultural Researcher checks the accuracy of the product increment.
4. The product increment is deployed to the Staging Environment.
5. The product increment is documented.

2.1.5.3 Definition of Ready

Definition of ready is the list of requirements that a user story must fulfill in order to be incorporated into the next sprint. The definition of ready for user stories was as follows:

1. The user story is presented in the meta language format: As a [type of user], I want [some goal], so that [some benefit/reason] is realized.
2. The user story conforms with the product vision.
3. The user story is independent, valuable, estimable, small and testable (INVEST).

4. The development team has estimated the user story.

2.1.6 Conduct Release Planning

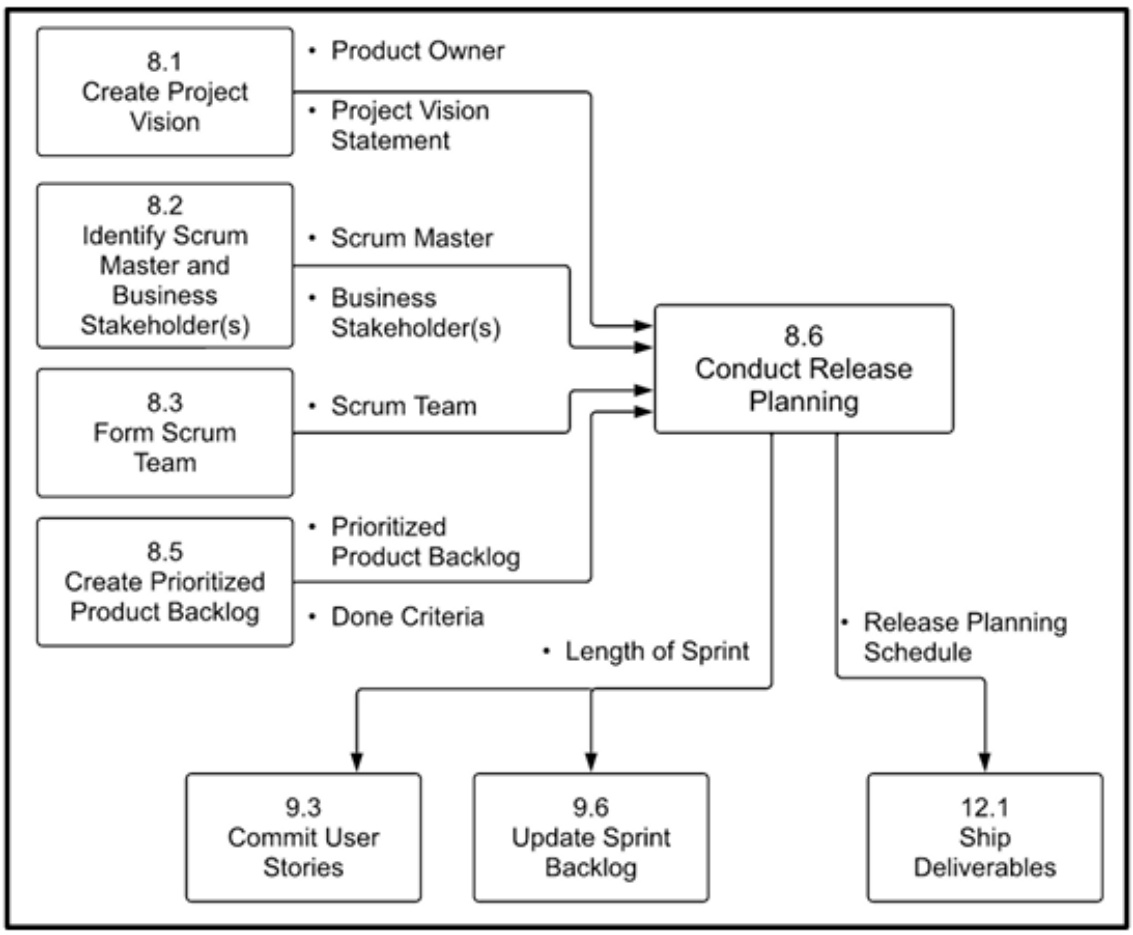


Figure 6. Data Flow Diagram for Conduct Release Planning Process (Satpathy, 2025)

2.1.6.1 Release Planning Schedule

Releasing Planning Schedule is the description of increments along with their release dates that is shared to the final users or internal team. A standard release contains a set of user stories (an increment) in the Prioritized Product Backlog, which are delivered and shipped together.

In our project, the Scrum team released increments every week after completion of Sprint 2.

Table 5. Release Planning Schedule

Release Name	**Release Date**	**Release Goal**	**Audience**	**Release Scope**
Release 0.1 (Alpha)	Dec 15, 2025 (End of Sprint 2)	Functional AI Model	Internal QA & PO	- Validated Image Dataset - Trained CNN Model - Accuracy Validation
Release 0.5 (Beta)	Dec 22, 2025 (End of Sprint 3)	Connected System	5 Regional Officers	- API Development - Cloud Integration -Prototype Interface
Release 1.0 (Gold)	Dec 29, 2025 (End of Sprint 4)	User-Ready App	All Users	- Refined Mobile UI - Dashboard - Security Config

2.2 Plan & Estimate

2.2.1 Create User Stories

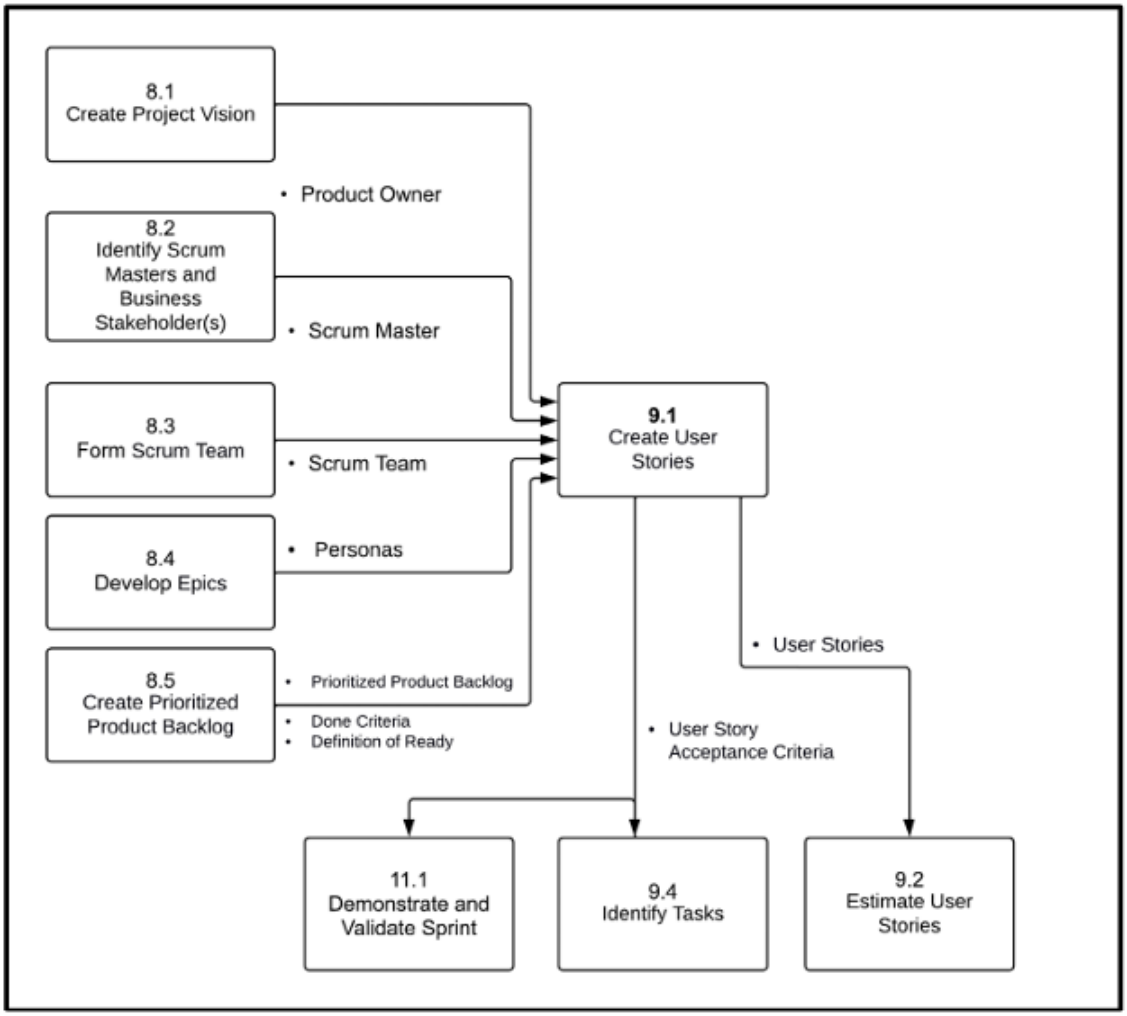


Figure 7. Data Flow Diagram for Create User Stories Process (Satpathy, 2025)

2.2.1.1 User Stories

User stories are the requirements of the users or customers of a software product. They are statements that are accurate and explain what and why the customer wants. User stories enable the development team to understand the value and the purpose of features they are developing. The User Stories of the MRWO Plant Doctor were as follows:

1. **MA-2:** As an Agricultural Researcher, I want to collect a varied and representative set of plant images so that the model can learn from real-world variations.
2. **MA-5:** As an Agricultural Researcher, I want to correctly label and organize collected images so that they can be used effectively for model training.
3. **MA-8:** As a Software Engineer, I want to preprocess image data so that the dataset is clean and suitable for model training.
4. **MA-11:** As a Software Engineer, I want to develop and evaluate a CNN model so that the system can accurately identify plant diseases.
5. **MA-17:** As a Backend Developer, I want to design APIs that receive image data and interact with the trained AI model so that the system can provide diagnoses.
6. **MA-20:** As a Backend Developer, I want to optimize API performance so that diagnosis requests are processed efficiently under higher loads.
7. **MA-25:** As a Farm Manager, I want a simple User Interface (UI) to upload images and view diagnosis results so that I can make informed decisions.
8. **MA-28:** As a Farm Manager, I want an online dashboard reporting disease trends so that my team can monitor crop health remotely.

2.2.1.2 User Stories Acceptance Criteria

User stories acceptance criteria are defined as conditions that a user story has to meet so that it can be accepted by the end user or customer. All user stories were accepted based on the following criteria:

- **MA-2: Data Collection**
    - **Quantity Target:** Collect 500+ “Ground Truth” images/captures from each of the 6 target categories (Wheat: Healthy, Rust, Leaf Spot; Cotton: Healthy, CLCV, Bacterial Blight).
    - **Environmental Variability:** Dataset must include samples taken under varying light conditions (bright and dark) to be reliable in varying situations.

    - **Spatial Variability:** Images must represent multiple angles and varying distances in order to replicate different styles of photography.

- **MA-5: Labeling & Organization**

    - **Classification Accuracy:** A random sample of the data must prove 100% labeling accuracy (e.g., an image of Rust will never be found in the Healthy folder).

    - **Structure Standardization:** The images must be arranged into a hierarchy of directories that is relevant to the needs of the training system.

    - **Format Consistency:** All files must be changed to a single file format (e.g., JPG) to avoid compatibility errors.

- **MA-8: Data Preprocessing**

    - **Resolution Normalization:** The input images must be automatically scaled to a predefined resolution that is required by the model (e.g. 224x224 pixels).

    - **Data Cleaning:** The system must automatically detect and delete corrupted or unreadable image files.

    - **Augmentation (Volume Increase):** The preprocessing pipeline must produce synthetic augmentations (rotations, flips, zooms) to augment the dataset by at least 3 times.

- **MA-11: CNN Model Development**

    - **Accuracy Threshold:** The model must have a validation accuracy of more than 90 percent after the training is finished.

    - **Class Coverage:** The model must be able to identify and distinguish between 6 different target classes in the two crops: Wheat (Healthy, Rust, Leaf Spot) and Cotton (Healthy, CLCV, Bacterial Blight).

    - **Model Efficiency:** The AI model file must be as small as possible (less than 100MB) so that it can be run on a common hardware.

- **MA-17: API**

- **Input Compatibility:** The interface must be able to handle standard image formats transferred from the client's app.
- **Response Structure:** The system must answer with a structured response: the name of the identified disease and a confidence score.
- **Offline Data Synchronization:** The API must support batch processing of queued images that were uploaded when the device was offline, returning the diagnosis results through a notification once the device's connectivity returns.
- **Error Handling:** The system must reject invalid inputs (e.g. text files or non-image data) by responding with a clear error message.

- **MA-20: API Optimization**
  - **Latency Target:** The time to process an image and provide a diagnosis must be less than 2 seconds.
  - **Concurrency:** The system must be able to support several concurrent requests (e.g. 10 users at the same time) without crashing.
  - **Memory Stability:** The system memory usage must be stable when the traffic is high.
- **MA-25: User Interface**
  - **Capture Mechanism:** The interface must allow the user to upload pictures both through the live camera and the device gallery.
  - **Result Visualization:** Results screen must clearly present the diagnosis (e.g. rust detected) with the confidence score.
  - **Usability:** The interface must adjust to fit various mobile screen sizes.
- **MA-28: Disease Trends Dashboard**
  - **Trend Visualization:** The dashboard should present graphical charts indicating the frequency of every disease over time.
  - **Filtering:** The users should be able to filter the information by date (e.g. Last 10 Days).

- **Access Control:** The dashboard should ensure secure access to prevent unauthorized personnel accessing the data.

2.2.2 Estimate User Stories

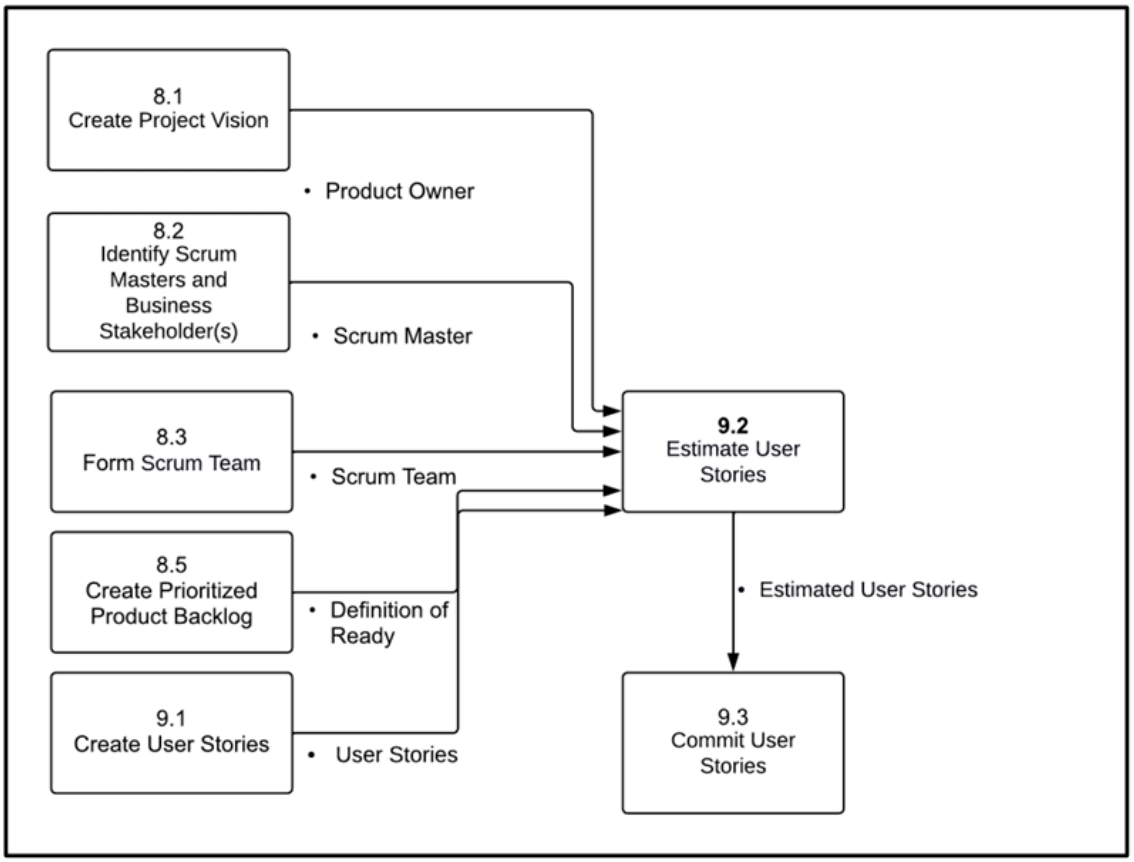


Figure 8. Data Flow Diagram for Estimate User Stories Process (Satpathy, 2025)

The Scrum team estimated user stories at every Sprint Planning meeting. The team applied a mutually agreed scale to determine the level of difficulty or effort to be applied to each user story. The Project estimation technique applied to estimate user stories was the Fixed-Point Method (20 Points) based on consistent delivery standards.

Table 6. User Story Estimation

Jira ID	**User Story Title**	**Estimate (Points)**
MA-2	Image Collection	20
MA-5	Labeling	20
MA-8	Preprocessing	20

MA-11	Train CNN Model	20
MA-17	Build API	20
MA-20	Optimize API	20
MA-25	Build UI	20
MA-28	Dashboard	20

2.2.3 Commit User Stories

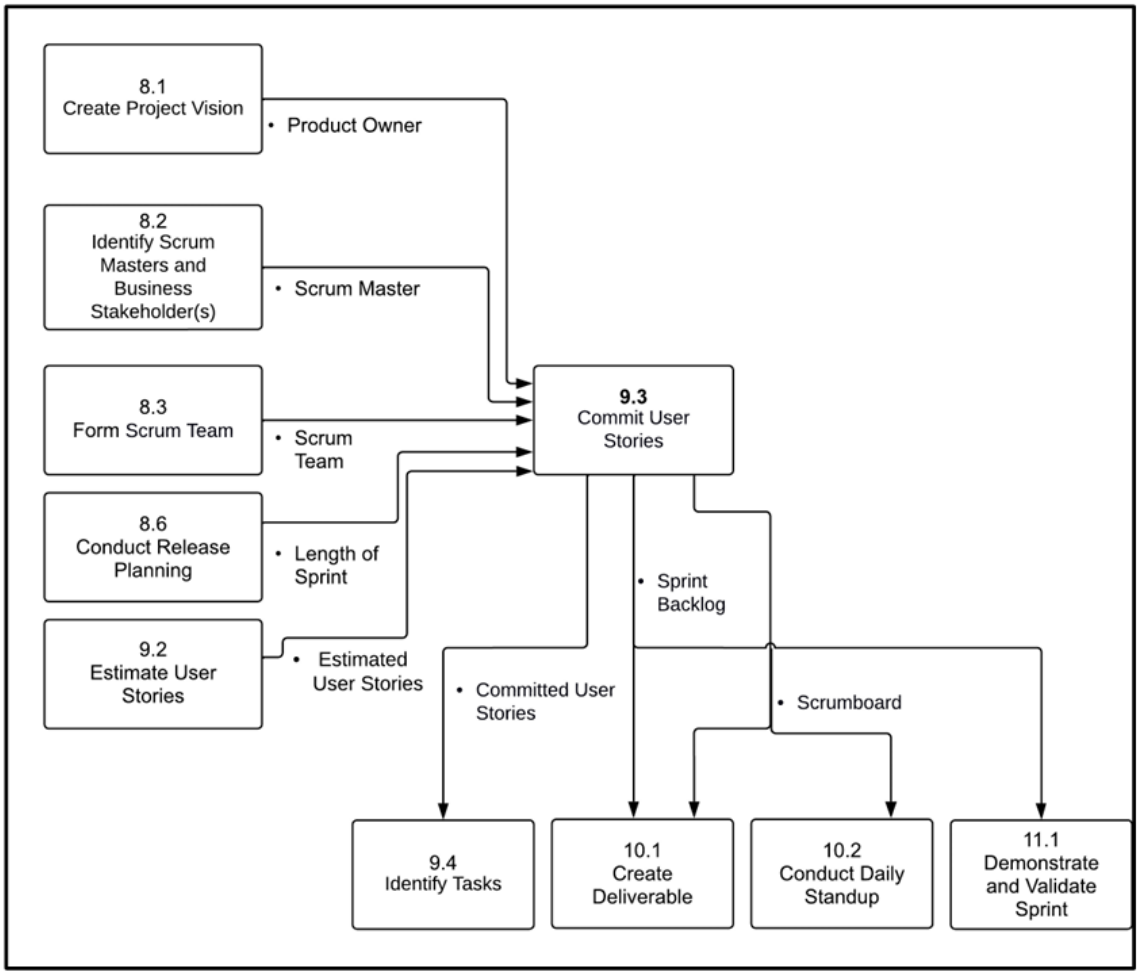


Figure 9. Data Flow Diagram for Commit User Stories Process (Satpathy, 2025)

2.2.3.1 Committed User Stories & Velocity Justification

The scrum team committed to user stories by choosing those with highest priority and those that could be completed in the next sprint.

Velocity Justification: The team chose a standard velocity using available working days in every sprint:

- **Standard Velocity (40 Points):** Within the first 4 days of the work week, the team was dedicated to 2 Major User Stories (2 stories x 20 points = 40 points).
- **Extended Velocity (50 Points):** For sprints that included critical bugs (Sprint 2 and 3), the team used the entire working week. This increased capacity allowed for an additional 10 Points allocated to Bug fixing.

Sprint 1 Commit (Total: 40 Points - 4 Days):

- MA-2: Image Collection (20 pts)
- MA-5: Labeling (20 pts)

Sprint 2 Commit (Total: 50 Points - 5 Days):

- MA-8: Preprocessing (20 pts)
- MA-11: Model Training (20 pts)
- Bug Fix MA-15: GPU Memory Overflow Resolution (10 pts)

Sprint 3 Commit (Total: 50 Points - 5 Days):

- MA-17: Build API & Cloud Setup (20 pts)
- MA-20: Optimize API Performance (20 pts)
- Bug Fix MA-24: Latency Resolution (10 pts)

Sprint 4 Commit (Total: 40 Points - 4 Days):

- MA-25: Build UI (20 pts)
- MA-28: Dashboard (20 pts)

2.2.3.2 Sprint Backlog

The Sprint Backlog is the chosen list of the User Stories and Tasks that will be completed by the Development/Scrum Team during the Sprint. These work items are drawn out of the Prioritized Product Backlog to be implemented in a specific Sprint.

2.2.3.3 Scrum board

The Scrum board visualizes workflow with columns: To Do, In Progress, Done.

Figure 10. Jira Scrum Board

2.2.4 Identify Tasks

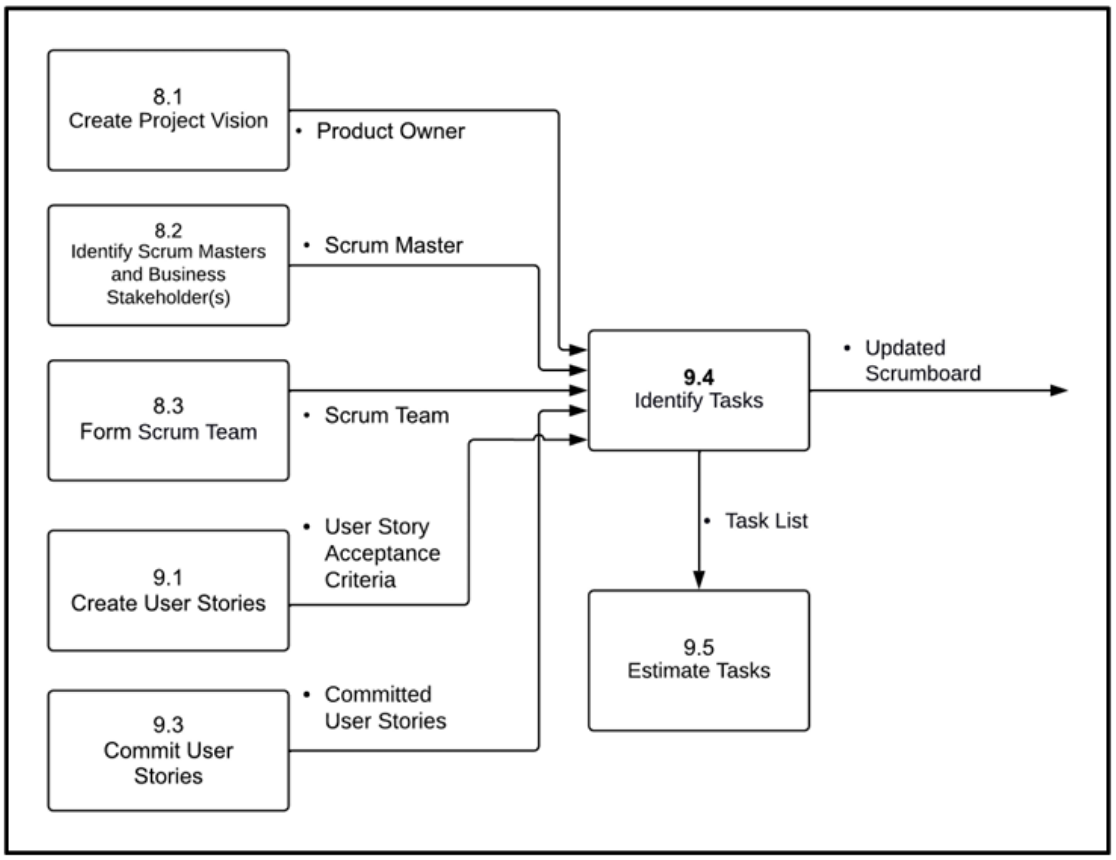


Figure 11. Data Flow Diagram for Identify Tasks Process (Satpathy, 2025)

Each user story included 2 technical tasks. (See Section 3.3 Product Backlog Status.)

2.2.5 Estimate Tasks

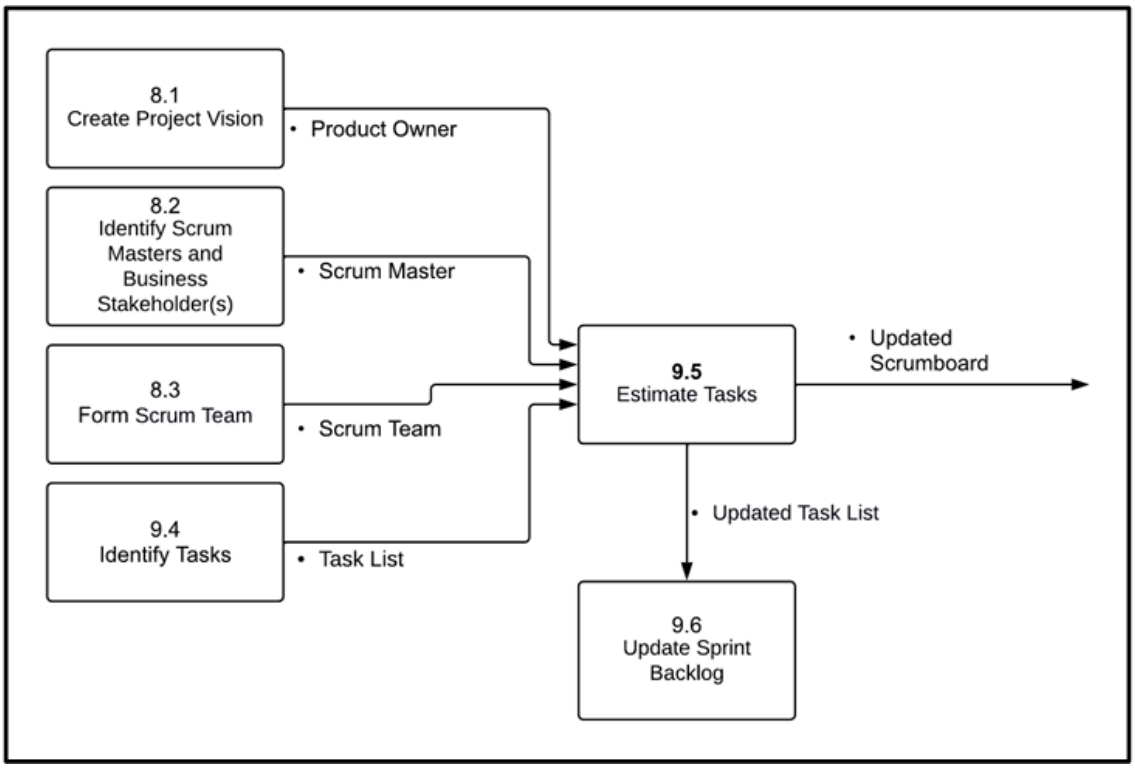


Figure 12. Data Flow Diagram for Estimate Tasks Process (Satpathy, 2025)

In order to simplify the tracking process and administration of the project, the team used a daily execution model to estimate tasks. Since each User Story included two technical tasks, the team was capable of completing one User Story in two day (where each activity consumed about one full working day).

Instead of granular hourly tracking, the team treated each task as a "day unit." This allowed for clearer goals during Daily Standups - either a task was "Done" by the end of the day or it was a blocker.

2.2.6 Update Sprint Backlog

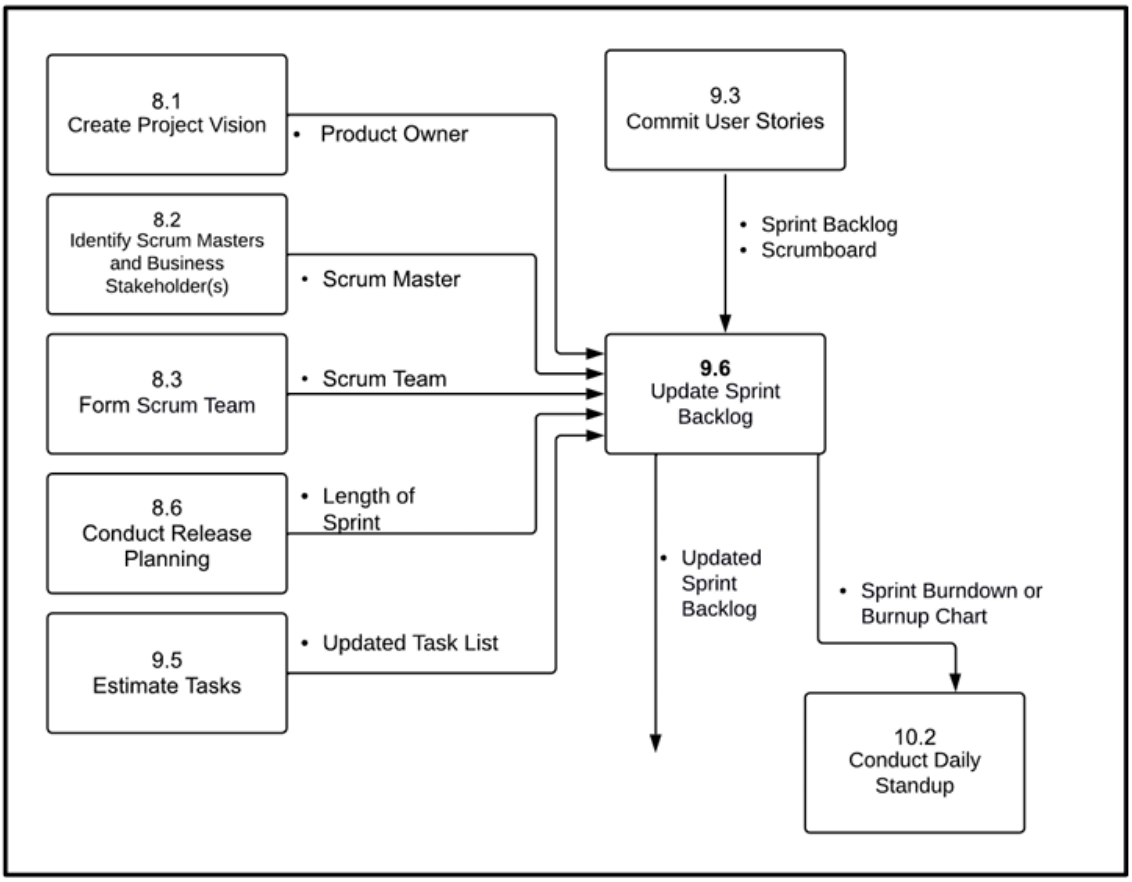


Figure 13. Data Flow Diagram for Update Sprint Backlog Process (Satpathy, 2025)

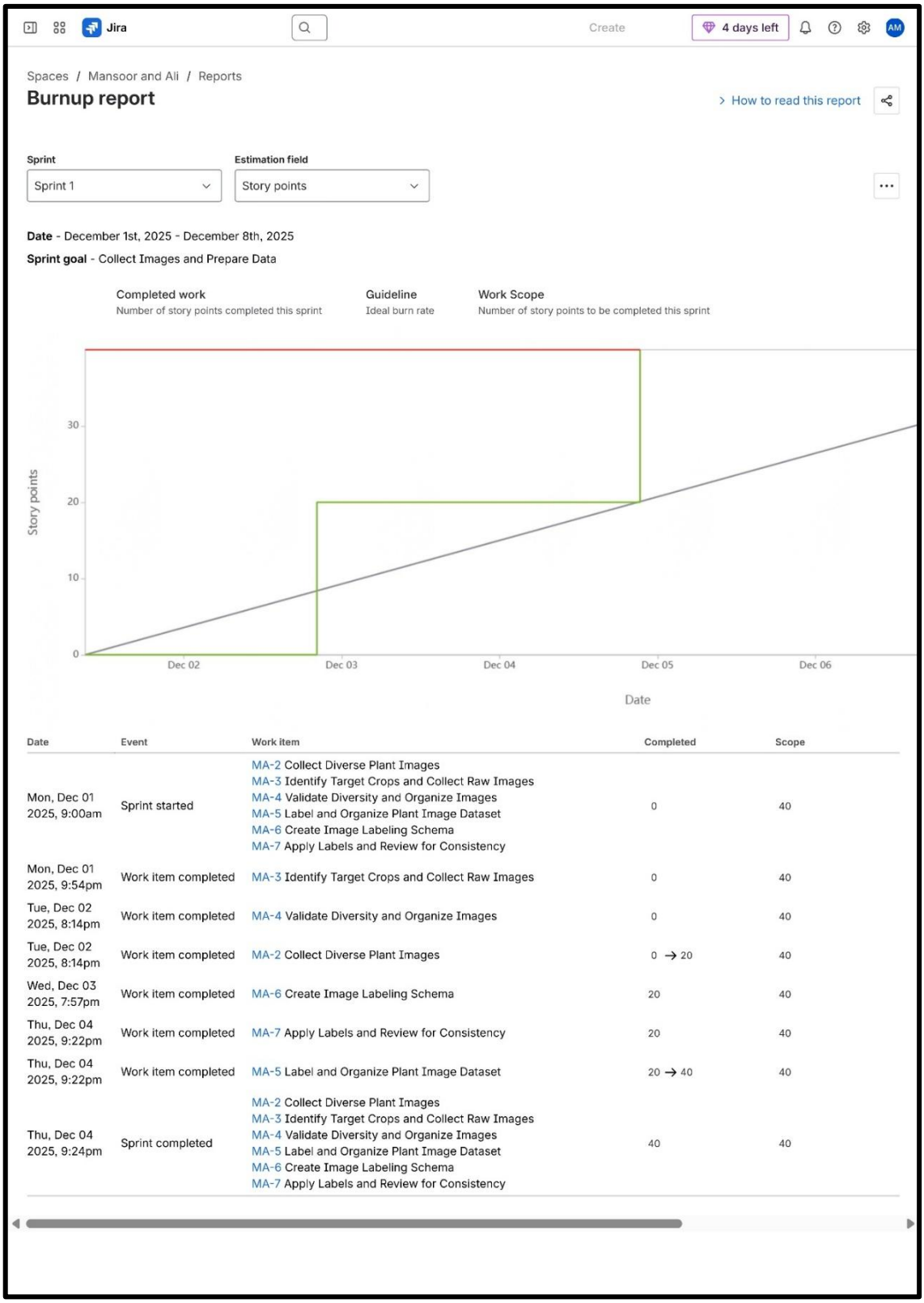
The Sprint Backlog was updated on a daily basis. In Sprints 2 and 3, 10 points (40 to 50) were added to the backlog to reflect the emergent bugs (MA-15 and MA-24) and, hence, the team had to use the 5th working day during those sprints.

2.2.6.3 Sprint Burn Up and Burn Down Charts

These charts tracked our progress.

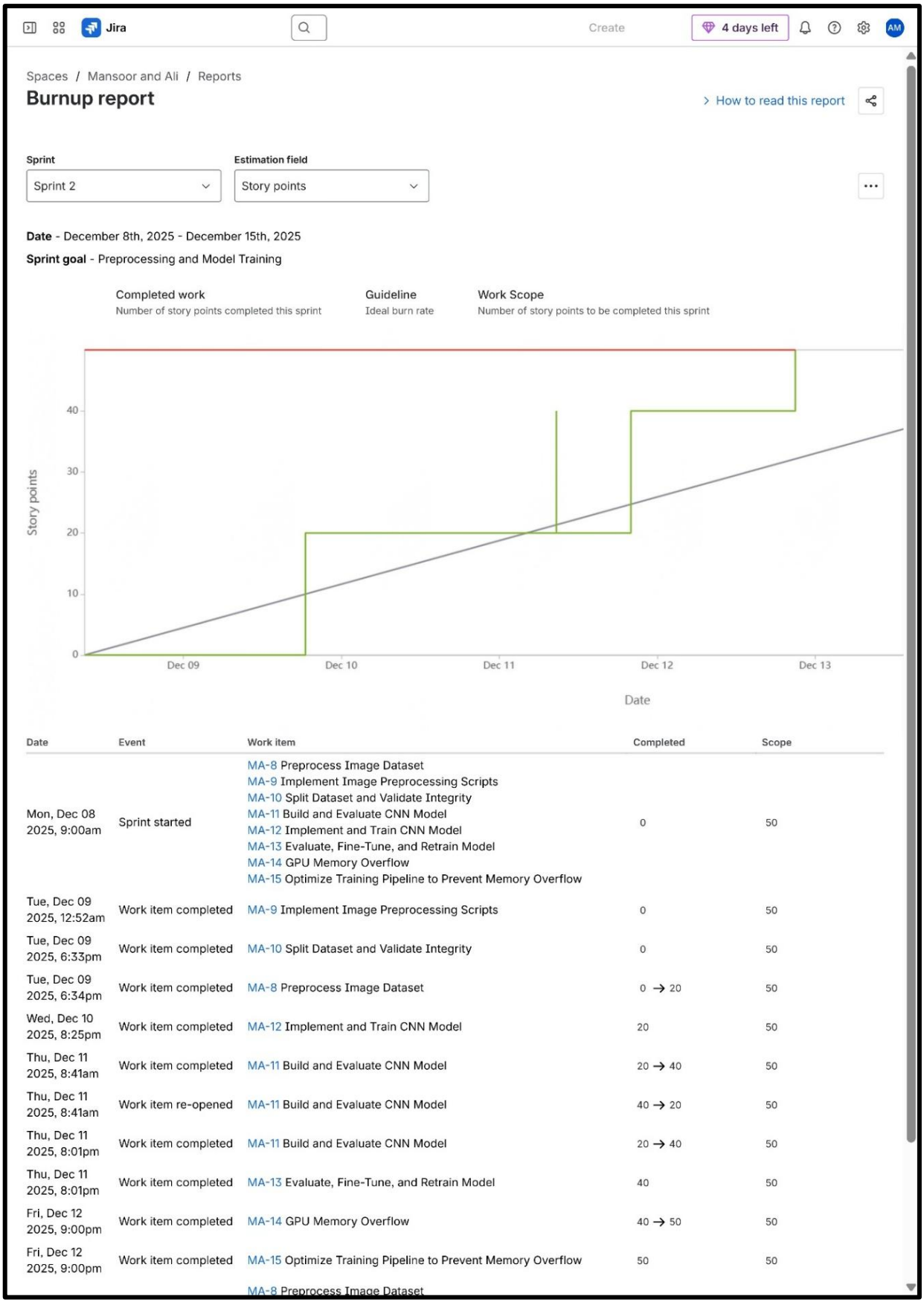
The vertical spikes in the Sprint 2 Burn Up and Burn Down charts were due to an administrative error and not an actual change of scope in the project. During a daily update of the Scrum board, a team member accidentally transferred a group of tasks between To Do and In Progress and, later, transferred them back to their correct workflow state. Unfortunately, this movement was recorded in the system as a noticeable variation in the

effort that day. This was just a data-entry error and did not indicate an actual shift in the workload or capacity of the team to complete the sprint deadline.



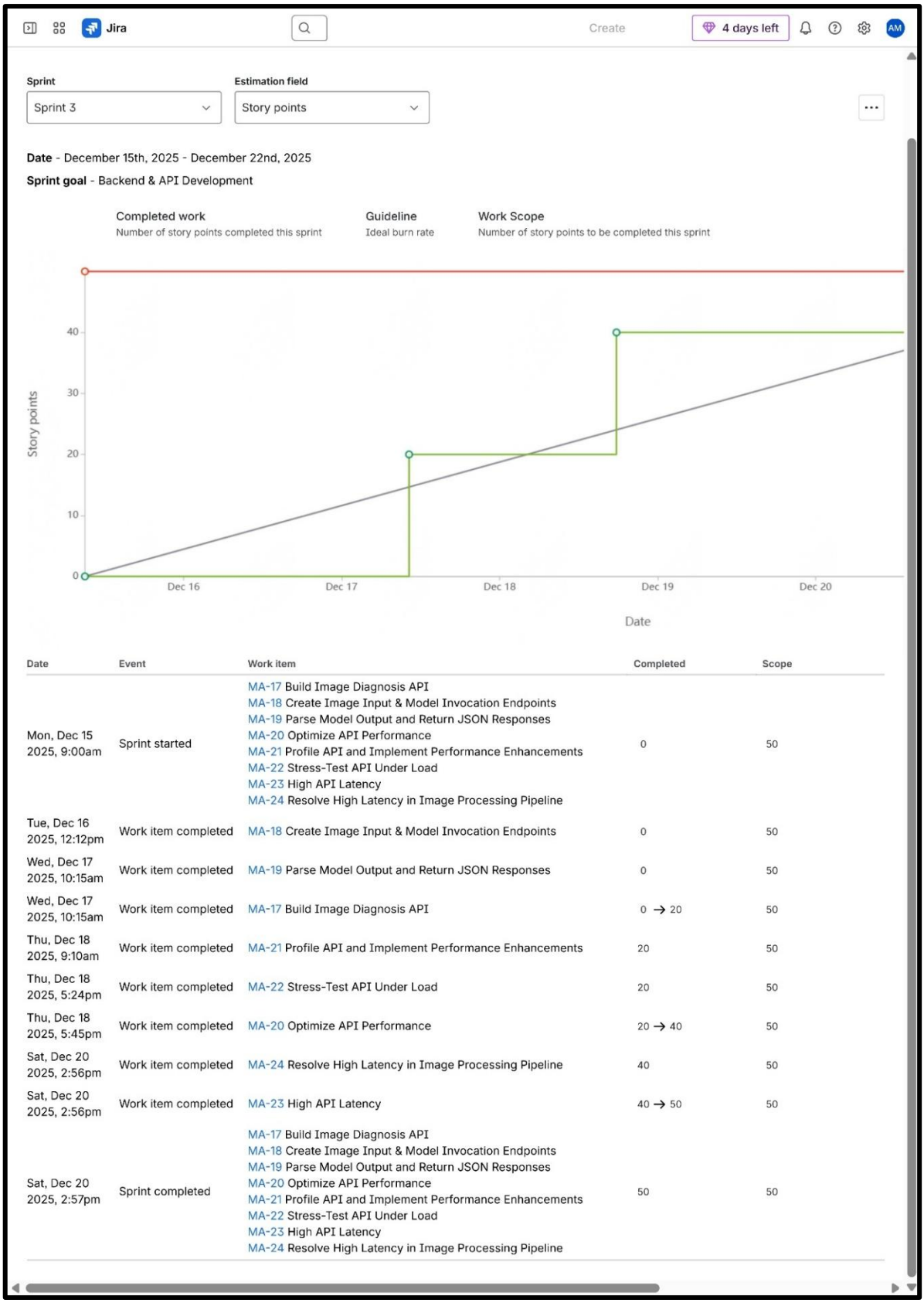
Date	Event	Work item	Completed	Scope
Mon, Dec 01 2025, 9:00am	Sprint started	MA-2 Collect Diverse Plant Images MA-3 Identify Target Crops and Collect Raw Images MA-4 Validate Diversity and Organize Images MA-5 Label and Organize Plant Image Dataset MA-6 Create Image Labeling Schema MA-7 Apply Labels and Review for Consistency	0	40
Mon, Dec 01 2025, 9:54pm	Work item completed	MA-3 Identify Target Crops and Collect Raw Images	0	40
Tue, Dec 02 2025, 8:14pm	Work item completed	MA-4 Validate Diversity and Organize Images	0	40
Tue, Dec 02 2025, 8:14pm	Work item completed	MA-2 Collect Diverse Plant Images	0 → 20	40
Wed, Dec 03 2025, 7:57pm	Work item completed	MA-6 Create Image Labeling Schema	20	40
Thu, Dec 04 2025, 9:22pm	Work item completed	MA-7 Apply Labels and Review for Consistency	20	40
Thu, Dec 04 2025, 9:22pm	Work item completed	MA-5 Label and Organize Plant Image Dataset	20 → 40	40
Thu, Dec 04 2025, 9:24pm	Sprint completed	MA-2 Collect Diverse Plant Images MA-3 Identify Target Crops and Collect Raw Images MA-4 Validate Diversity and Organize Images MA-5 Label and Organize Plant Image Dataset MA-6 Create Image Labeling Schema MA-7 Apply Labels and Review for Consistency	40	40

Figure 14. Sprint 1 Burnup Chart



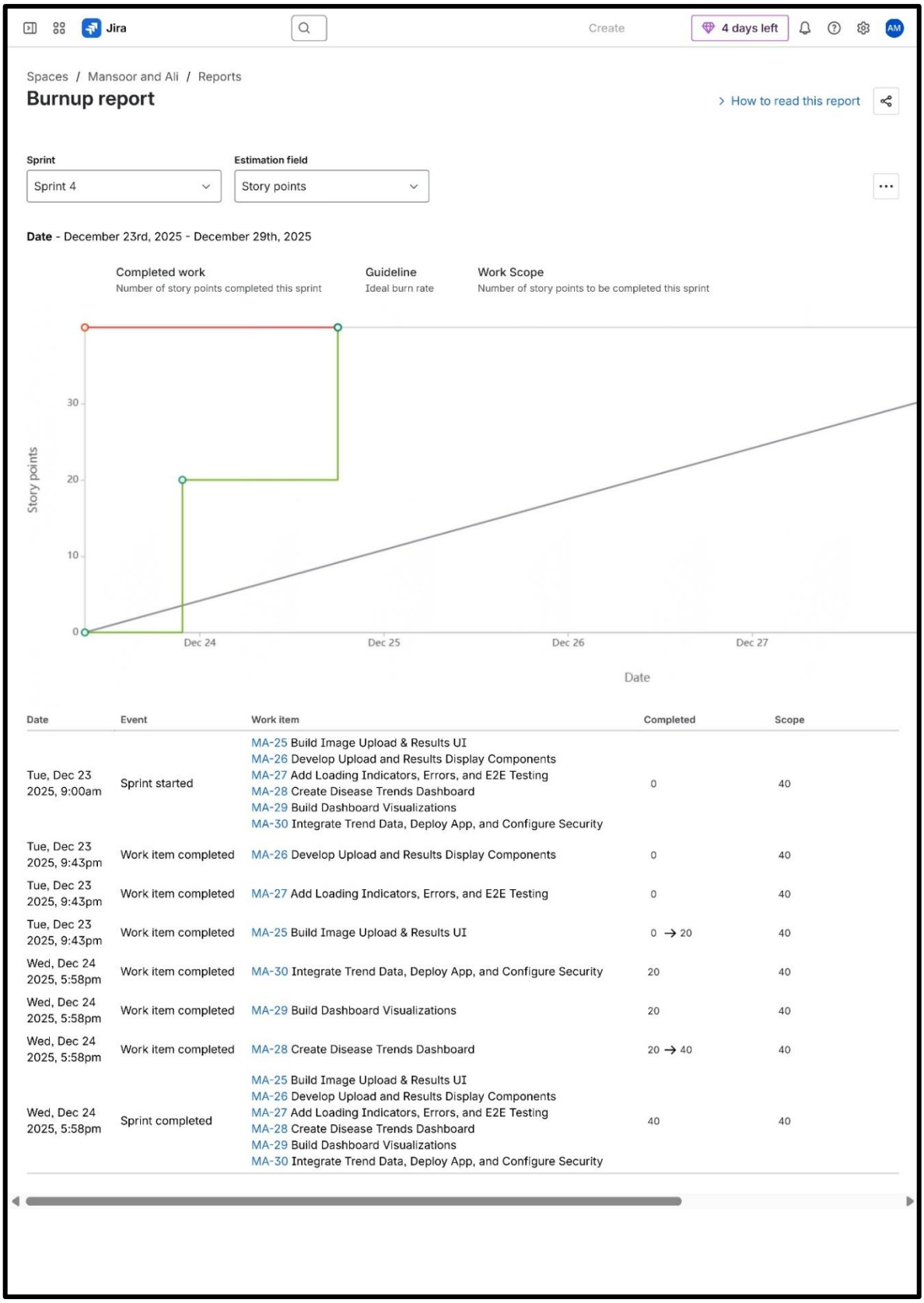
Date	Event	Work item	Completed	Scope
Mon, Dec 08 2025, 9:00am	Sprint started	MA-8 Preprocess Image Dataset MA-9 Implement Image Preprocessing Scripts MA-10 Split Dataset and Validate Integrity MA-11 Build and Evaluate CNN Model MA-12 Implement and Train CNN Model MA-13 Evaluate, Fine-Tune, and Retrain Model MA-14 GPU Memory Overflow MA-15 Optimize Training Pipeline to Prevent Memory Overflow	0	50
Tue, Dec 09 2025, 12:52am	Work item completed	MA-9 Implement Image Preprocessing Scripts	0	50
Tue, Dec 09 2025, 6:33pm	Work item completed	MA-10 Split Dataset and Validate Integrity	0	50
Tue, Dec 09 2025, 6:34pm	Work item completed	MA-8 Preprocess Image Dataset	0 → 20	50
Wed, Dec 10 2025, 8:25pm	Work item completed	MA-12 Implement and Train CNN Model	20	50
Thu, Dec 11 2025, 8:41am	Work item completed	MA-11 Build and Evaluate CNN Model	20 → 40	50
Thu, Dec 11 2025, 8:41am	Work item re-opened	MA-11 Build and Evaluate CNN Model	40 → 20	50
Thu, Dec 11 2025, 8:01pm	Work item completed	MA-11 Build and Evaluate CNN Model	20 → 40	50
Thu, Dec 11 2025, 8:01pm	Work item completed	MA-13 Evaluate, Fine-Tune, and Retrain Model	40	50
Fri, Dec 12 2025, 9:00pm	Work item completed	MA-14 GPU Memory Overflow	40 → 50	50
Fri, Dec 12 2025, 9:00pm	Work item completed	MA-15 Optimize Training Pipeline to Prevent Memory Overflow	50	50
		MA-8 Preprocess Image Dataset		

Figure 15. Sprint 2 Burnup Chart



Date	Event	Work item	Completed	Scope
Mon, Dec 15 2025, 9:00am	Sprint started	MA-17 Build Image Diagnosis API MA-18 Create Image Input & Model Invocation Endpoints MA-19 Parse Model Output and Return JSON Responses MA-20 Optimize API Performance MA-21 Profile API and Implement Performance Enhancements MA-22 Stress-Test API Under Load MA-23 High API Latency MA-24 Resolve High Latency in Image Processing Pipeline	0	50
Tue, Dec 16 2025, 12:12pm	Work item completed	MA-18 Create Image Input & Model Invocation Endpoints	0	50
Wed, Dec 17 2025, 10:15am	Work item completed	MA-19 Parse Model Output and Return JSON Responses	0	50
Wed, Dec 17 2025, 10:15am	Work item completed	MA-17 Build Image Diagnosis API	0 → 20	50
Thu, Dec 18 2025, 9:10am	Work item completed	MA-21 Profile API and Implement Performance Enhancements	20	50
Thu, Dec 18 2025, 5:24pm	Work item completed	MA-22 Stress-Test API Under Load	20	50
Thu, Dec 18 2025, 5:45pm	Work item completed	MA-20 Optimize API Performance	20 → 40	50
Sat, Dec 20 2025, 2:56pm	Work item completed	MA-24 Resolve High Latency in Image Processing Pipeline	40	50
Sat, Dec 20 2025, 2:56pm	Work item completed	MA-23 High API Latency	40 → 50	50
Sat, Dec 20 2025, 2:57pm	Sprint completed	MA-17 Build Image Diagnosis API MA-18 Create Image Input & Model Invocation Endpoints MA-19 Parse Model Output and Return JSON Responses MA-20 Optimize API Performance MA-21 Profile API and Implement Performance Enhancements MA-22 Stress-Test API Under Load MA-23 High API Latency MA-24 Resolve High Latency in Image Processing Pipeline	50	50

Figure 16. Sprint 3 Burnup Chart



Date	Event	Work item	Completed	Scope
Tue, Dec 23 2025, 9:00am	Sprint started	MA-25 Build Image Upload & Results UI MA-26 Develop Upload and Results Display Components MA-27 Add Loading Indicators, Errors, and E2E Testing MA-28 Create Disease Trends Dashboard MA-29 Build Dashboard Visualizations MA-30 Integrate Trend Data, Deploy App, and Configure Security	0	40
Tue, Dec 23 2025, 9:43pm	Work item completed	MA-26 Develop Upload and Results Display Components	0	40
Tue, Dec 23 2025, 9:43pm	Work item completed	MA-27 Add Loading Indicators, Errors, and E2E Testing	0	40
Tue, Dec 23 2025, 9:43pm	Work item completed	MA-25 Build Image Upload & Results UI	0 → 20	40
Wed, Dec 24 2025, 5:58pm	Work item completed	MA-30 Integrate Trend Data, Deploy App, and Configure Security	20	40
Wed, Dec 24 2025, 5:58pm	Work item completed	MA-29 Build Dashboard Visualizations	20	40
Wed, Dec 24 2025, 5:58pm	Work item completed	MA-28 Create Disease Trends Dashboard	20 → 40	40
Wed, Dec 24 2025, 5:58pm	Sprint completed	MA-25 Build Image Upload & Results UI MA-26 Develop Upload and Results Display Components MA-27 Add Loading Indicators, Errors, and E2E Testing MA-28 Create Disease Trends Dashboard MA-29 Build Dashboard Visualizations MA-30 Integrate Trend Data, Deploy App, and Configure Security	40	40

Figure 17. Sprint 4 Burnup Chart

Figure 18. Sprint 1 Burndown Chart

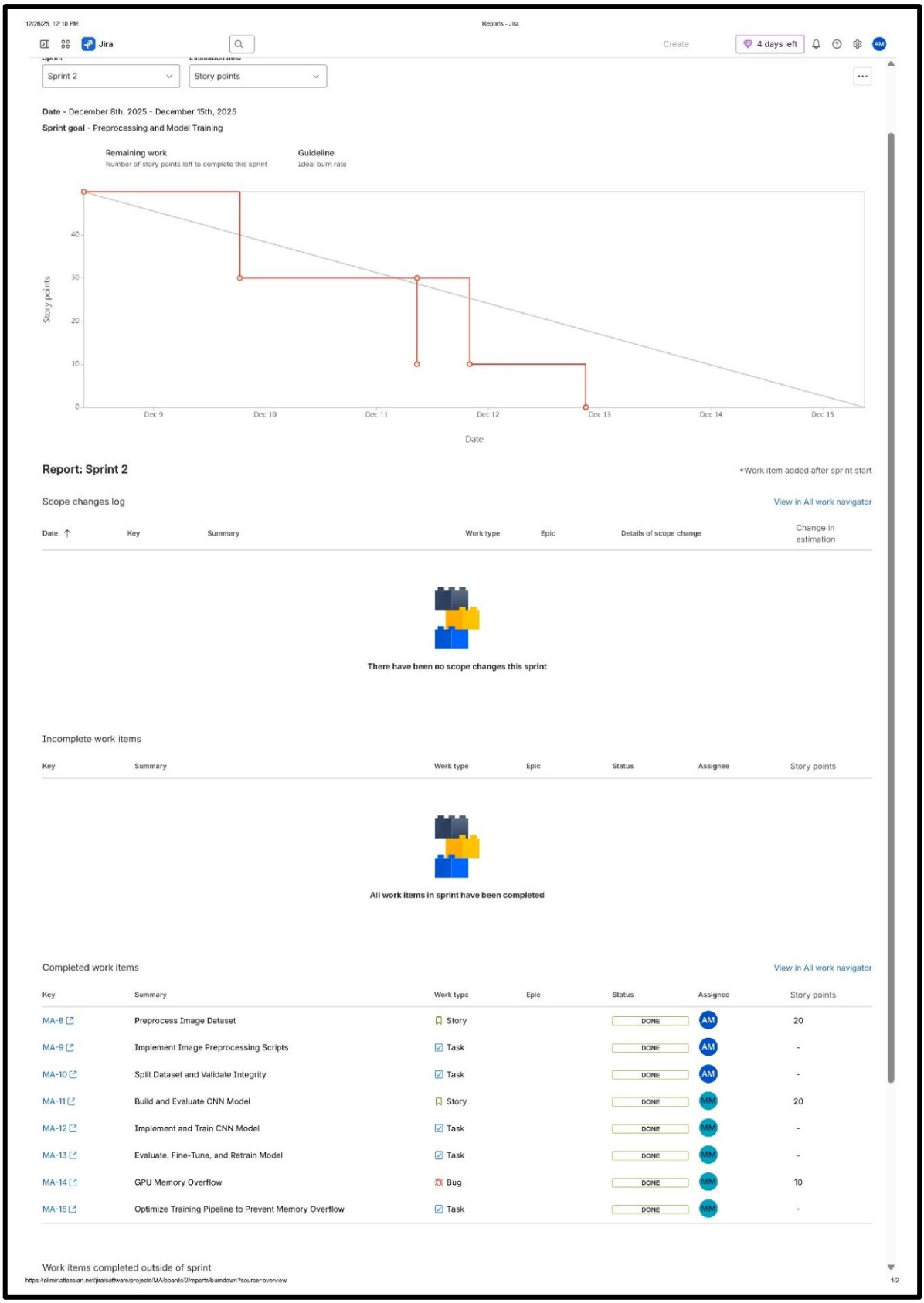


Figure 19. Sprint 2 Burndown Chart

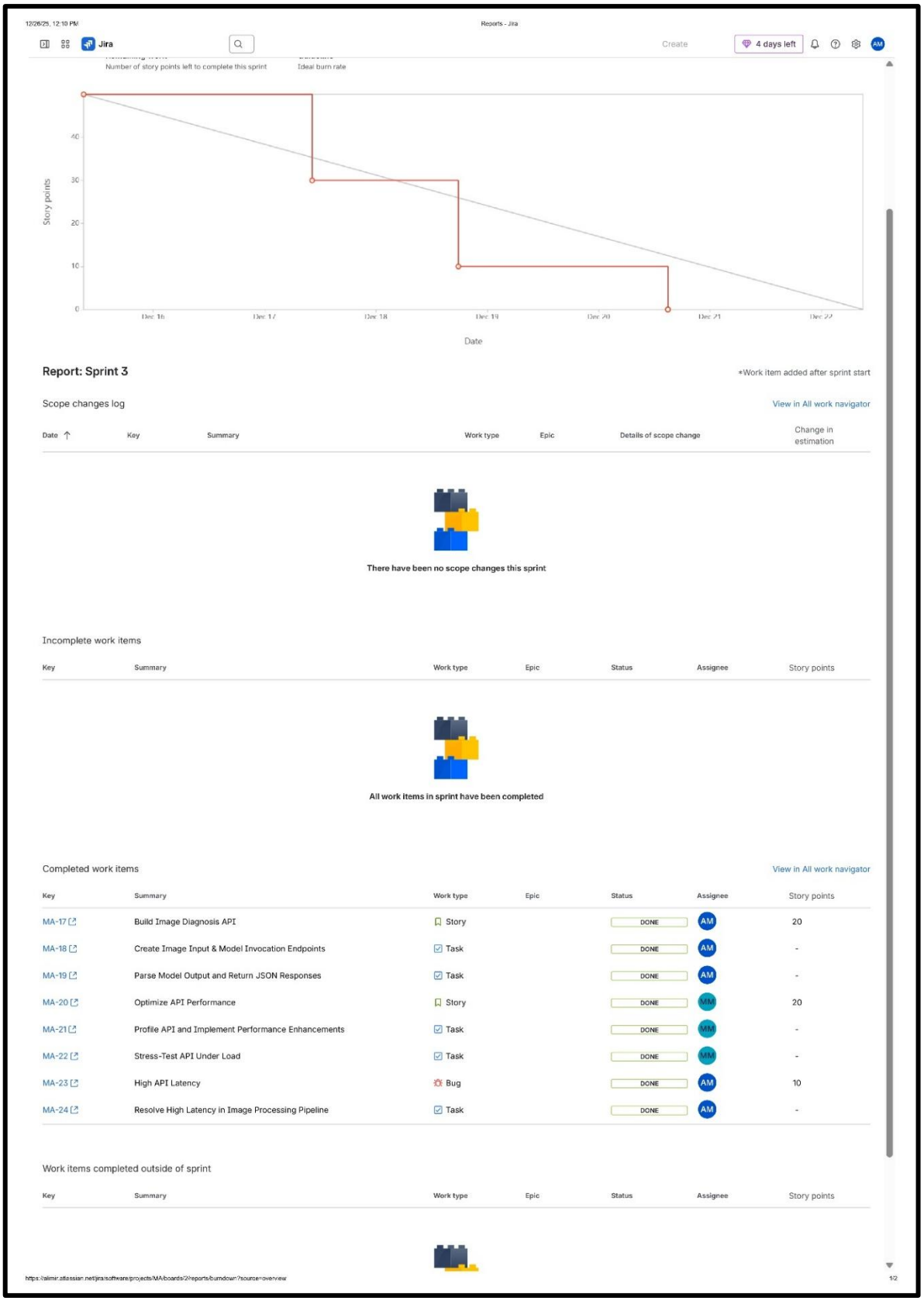


Figure 20. Sprint 3 Burndown Chart

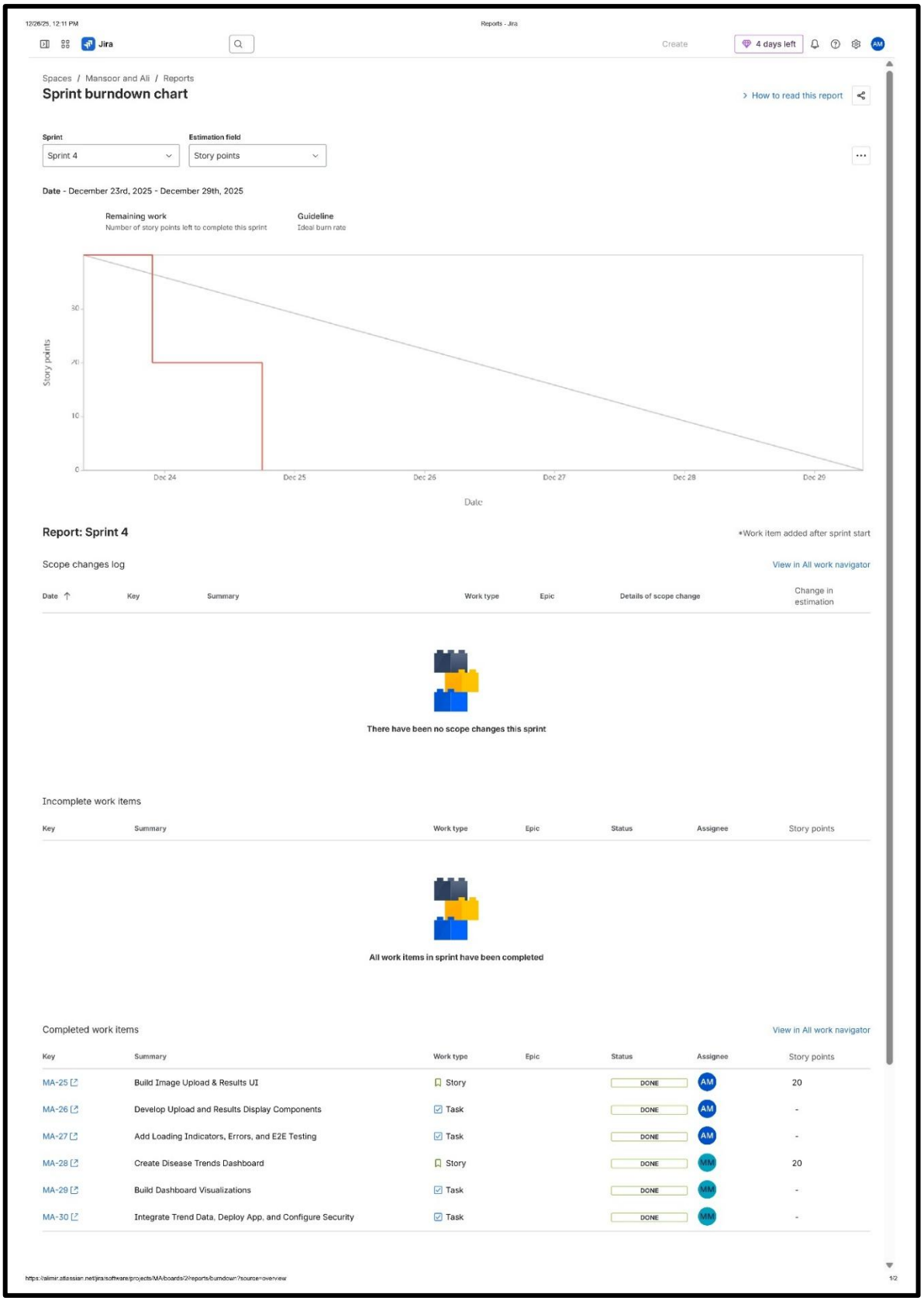


Figure 21. Sprint 4 Burndown Chart

2.3 Implement

2.3.1 Create Deliverables

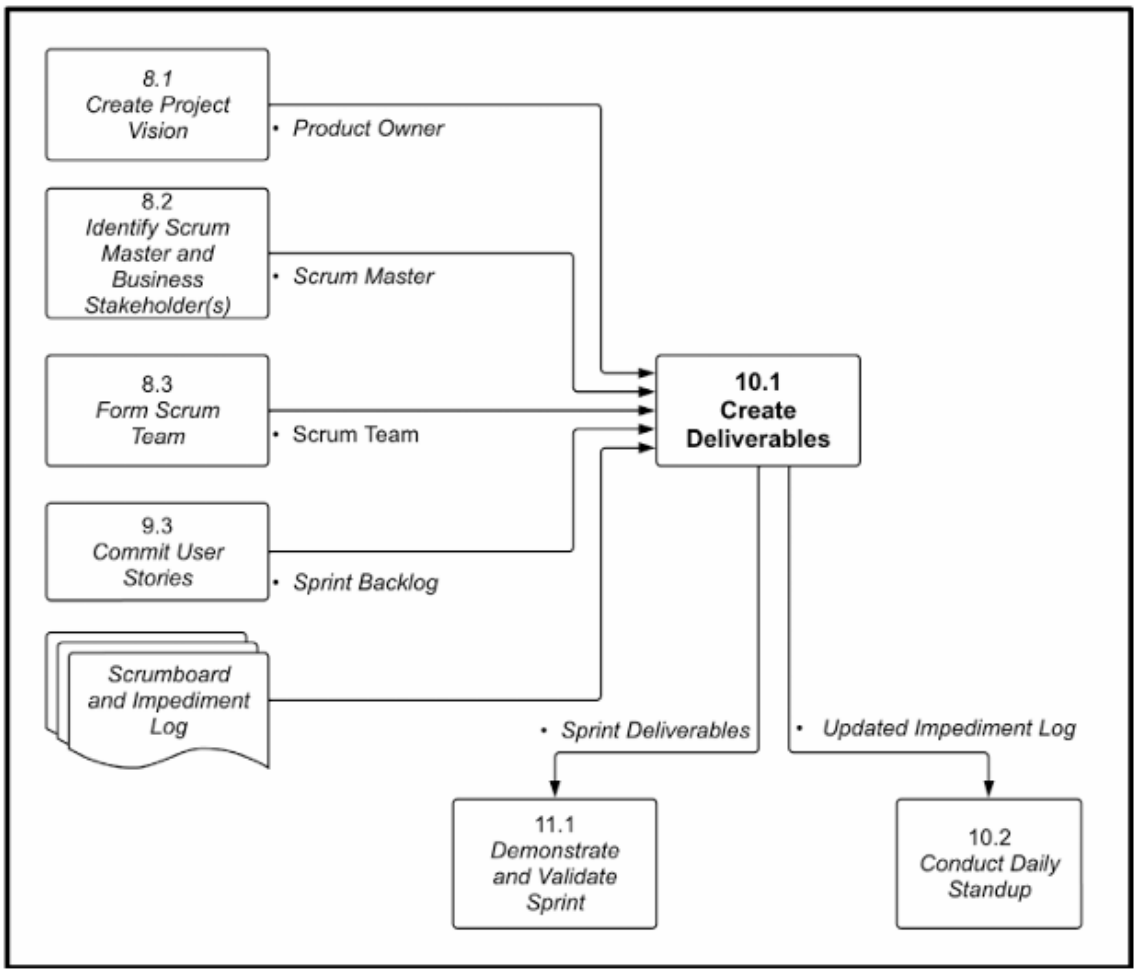


Figure 22. Data Flow Diagram for Create Deliverables Process (Satpathy, 2025)

2.3.1.1 Sprint Deliverables

- **Sprint 1:** A verified, labeled dataset of 3,000 images.
- **Sprint 2:** A trained AI model file.
- **Sprint 3:** A functional API endpoint on the cloud.
- **Sprint 4:** The Android APK file installed on farmer's phones.

2.3.2 Conduct Daily Standup

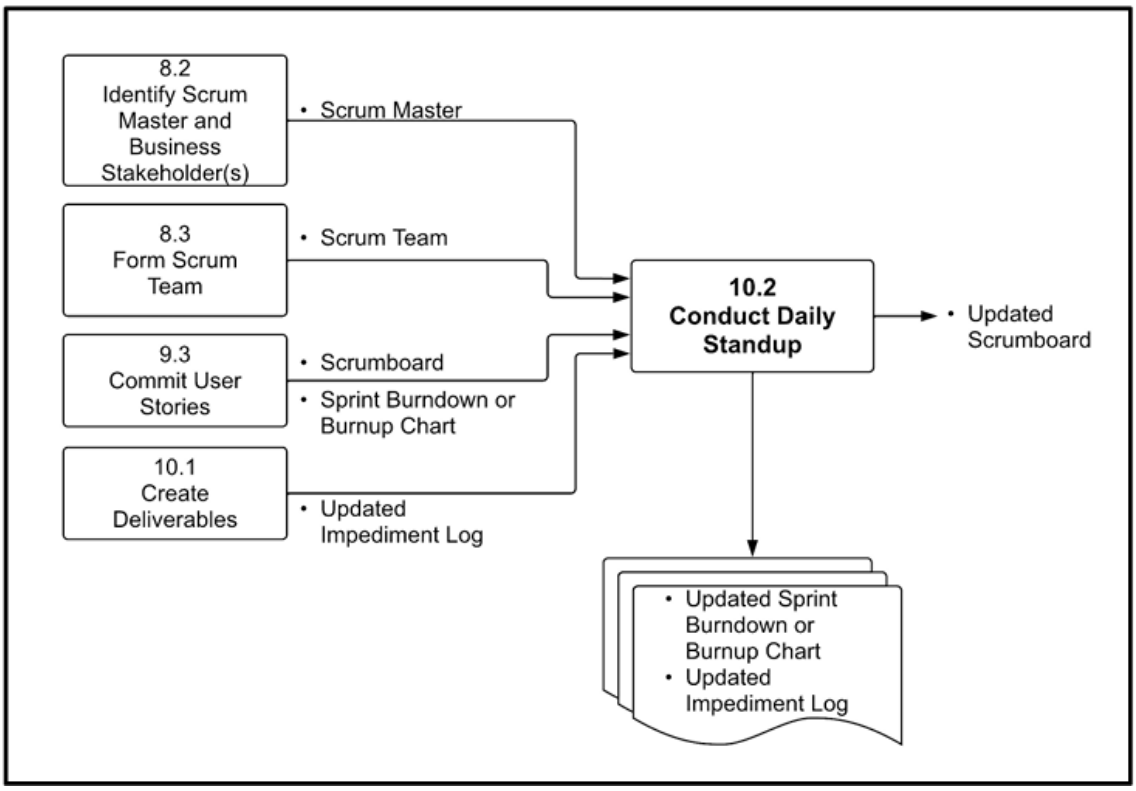


Figure 23. Data Flow Diagram for Conduct Daily Standup Process (Satpathy, 2025)

Daily Standup Log (Sample - Dec 10):

- **Scrum Master:** "Any blockers?"
- **Software Engineer:** "Yes. The GPU memory is overflowing during training (MA-14)."
- **Scrum Master:** I will report a Bug ticket. Can we reduce batch size?"
- **Agricultural Researcher:** "I am done with labeling the Wheat Rust pictures."

2.4 Review and Retrospect

2.4.1 Demonstrate and Validate Sprint

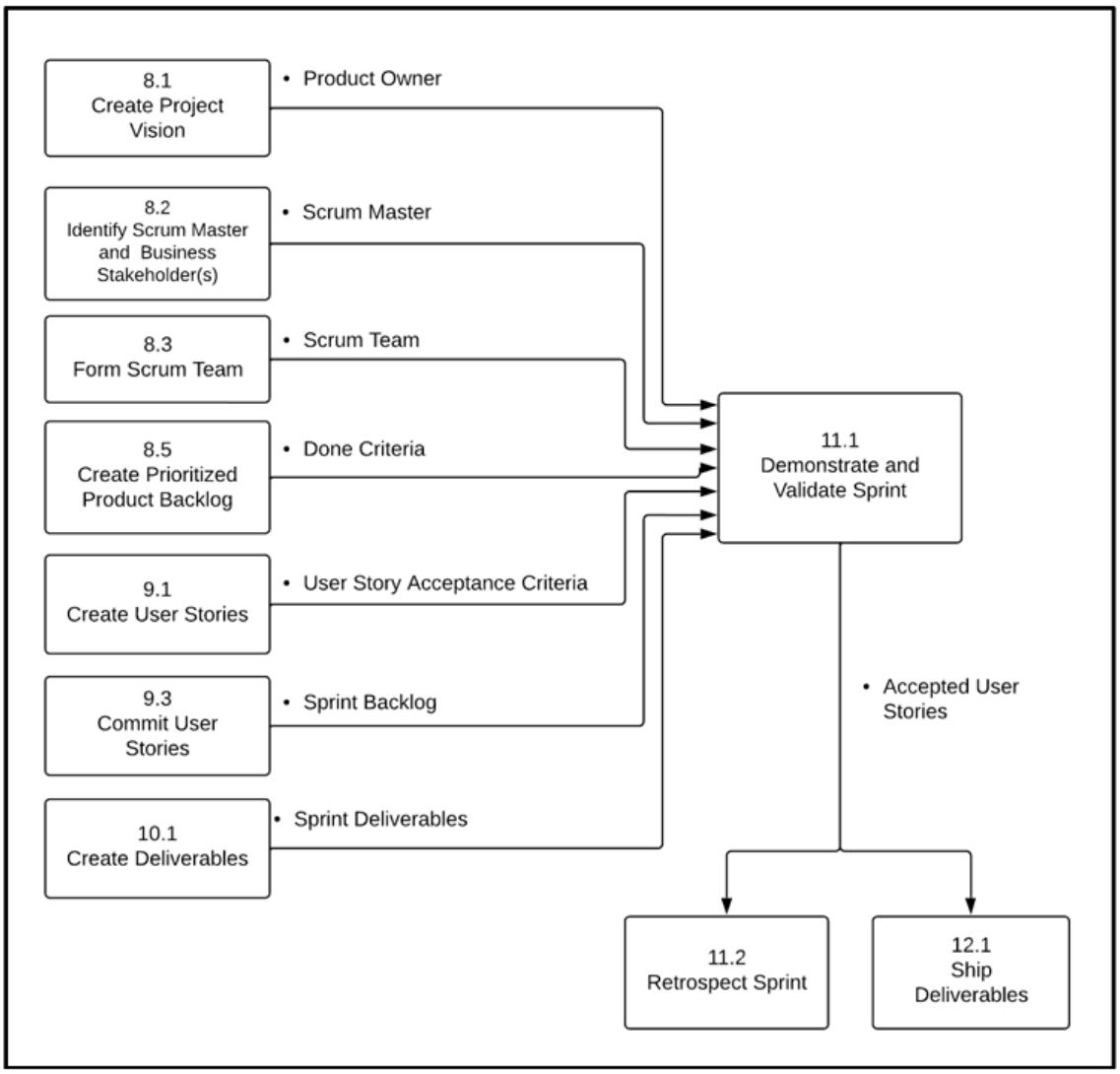


Figure 24. Data Flow Diagram for Demonstrate and Validate Sprint Process (Satpathy, 2025)

2.4.1.1 Accepted User Stories and Sprint Reviews

The deliverables were reviewed and accepted by the Product Owner at the end of every sprint, and they were found to be in line with the acceptance criteria.

- **Sprint 1 Review:** The "Validated Image Dataset" was accepted after the Product Owner verified the correct labeling of 3,000 images across the 6 target groups.
- **Sprint 2 Review:** "The Trained AI Model" was approved after validation against test data with over 92% accuracy. The team also fixed the GPU Memory Overflow Bug, ensuring stable training.
- **Sprint 3 Review:** The "Functional API Endpoint" was accepted. The team demonstrated successful API responses and the resolution of High Latency Bug, achieving response times under 2 seconds.
- **Sprint 4 Review:** The Product Owner reviewed the MRWO Plant Doctor app and Disease Trends Dashboard and approved them after confirming that the products worked on android smartphones.

2.4.2 Retrospect Sprint

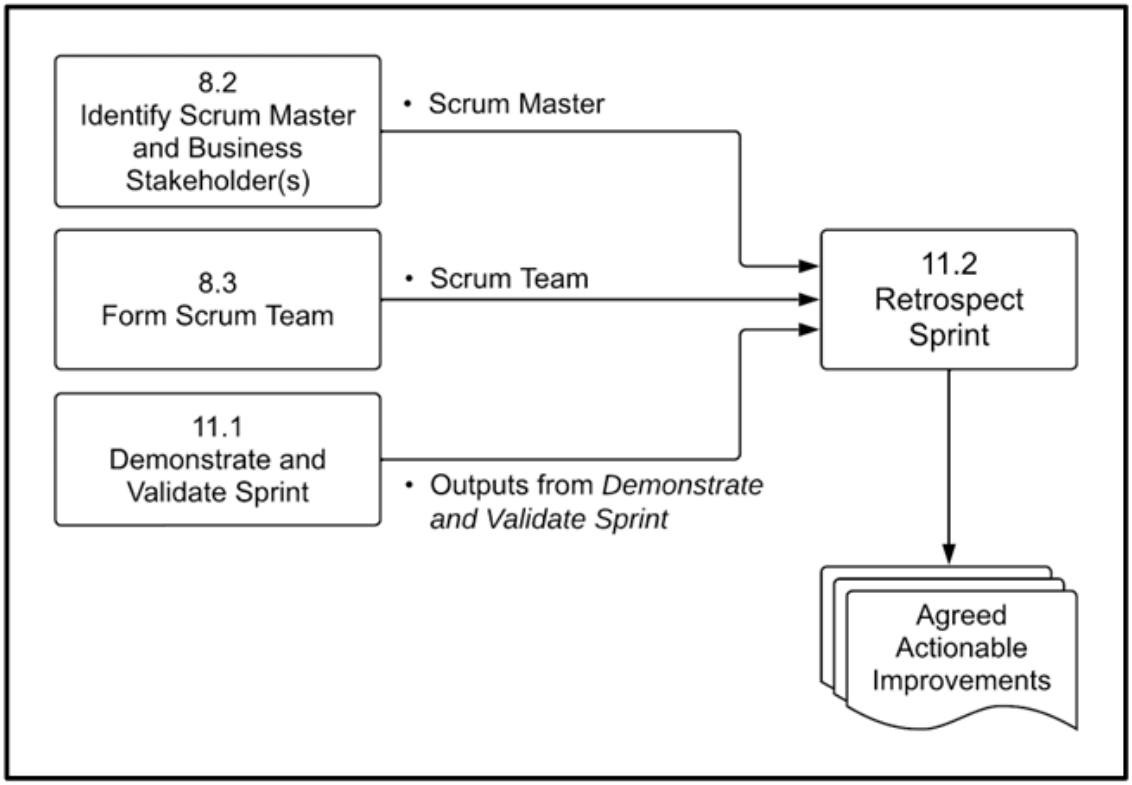


Figure 25. Data Flow Diagram for Retrospect Sprint Process (Satpathy, 2025)

Retrospective Findings:

Scrum Team held a retrospective at the end of every sprint to review their processes and make necessary changes on the spot.

- **Sprint 1 Retrospective:**
    - Finding: The collaboration between the Agricultural Researcher (Domain Expert) and Software Engineer was effective for data cleaning.
    - Action: The team confirmed that the baseline velocity of 40 Story Points was realistic for a standard 4-day working week.
- **Sprint 2 Retrospective:**
    - Finding: GPU Memory Overflow posed a threat to the sprint's progress. One of the bottlenecks was found to be local hardware.
    - Action: The team decided to extend the sprint's duration to 5 days and velocity to 50 Story Points. This enabled them complete the bug fix in Sprint 2 without transferring work to the next sprint.
- **Sprint 3 Retrospective:**
    - Finding: API Latency detection confirmed that the model was reloading from the cloud server on each request.
    - Action: The team held on to the 50-point Surge Velocity to implement the fixes (hot-loading and caching), which helped the system reach the 2-second response target.
- **Sprint 4 Retrospective:**
    - Finding: Field tests indicated that internet connectivity in Lodhran was unreliable, and training equipment in the area was not scalable.
    - Action: The team documented two major requirements for the next project: adding an offline mode feature for the app and budgeting for cloud GPUs for future model training.

2.5 Release

The Release stage involves the release of the final product increment to the client (or internal team) and identifying improvements for future projects. This phase marks the formal conclusion of the product development cycle.

2.5.1 Ship Deliverables

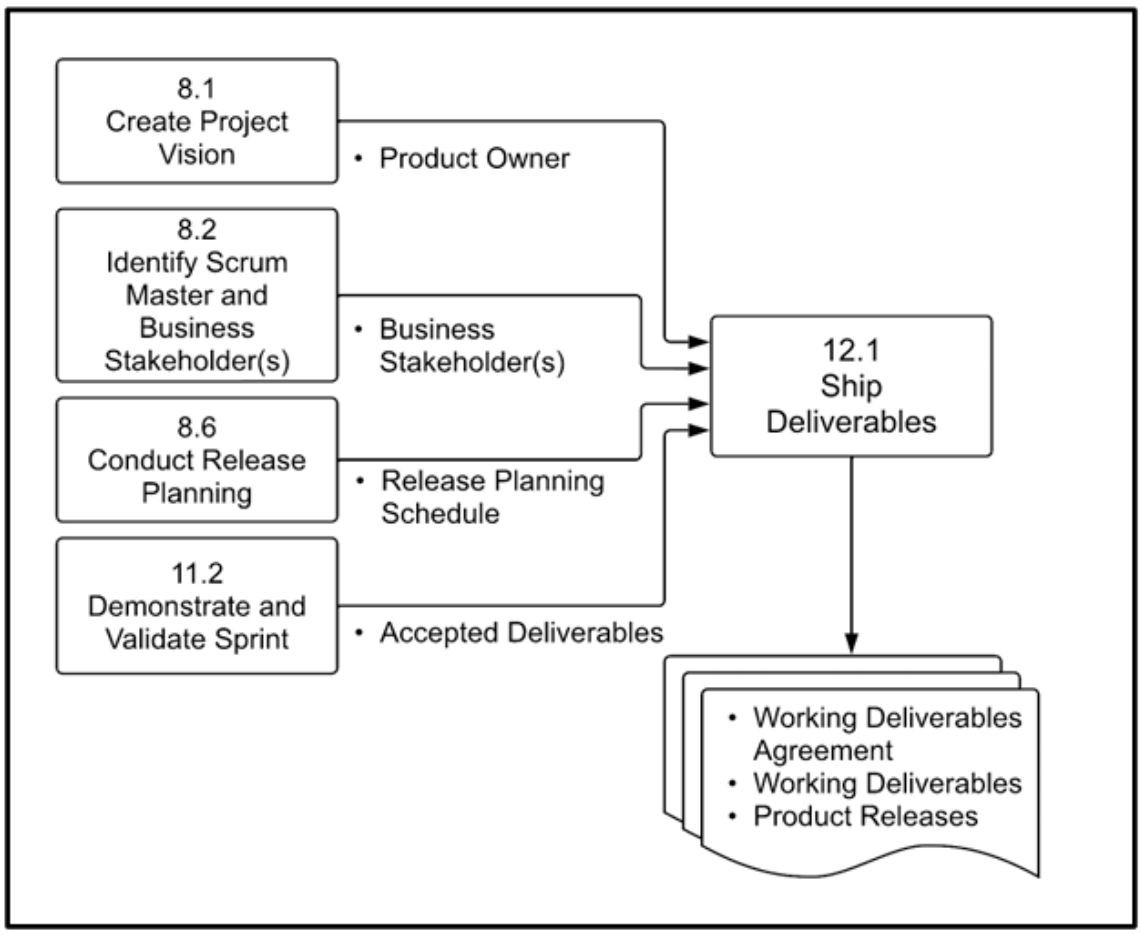


Figure 26. Data Flow Diagram for Ship Deliverables Process (Satpathy, 2025)

Table 7. Actual Increment Release Log

Increment Version	**Actual Release Date**	**Environment**	**Audience**	**Key Capabilities / Status**

v0.1 (Alpha)	Dec 15, 2025	Local Staging	Internal QA & PO	Delivered Validated Image Dataset & Functional AI Model via Command Line Interface (CLI).
v0.5 (Beta)	Dec 22, 2025	Pilot Devices	5 Regional Officers	Integrated the Backend API with a Prototype App.
v1.0 (Gold)	Dec 29, 2025	Field Operations	All Users	Final Release. Full UI, Dashboard and User Manual

2.5.1.1 Final Project Release

For the MRWO Plant Doctor project, the final release (Version 1.0) happened on December 29, 2025.

Final Release Components:

- **Android Application (.apk):** The app was sent to the Farm Managers and 50 pilot farmers in Lodhran through a direct secure download link to avoid Play Store delays.
- **Web Dashboard:** "Disease Trends Dashboard" was deployed to a secure server, which could only be accessed by MRWO Directors using VPN.
- **User Manual:** A comprehensive User Manual in Urdu and English was provided to the MRWO training team.

2.5.1.2 Organizational Readiness

To ensure a successful launch, the Scrum Team organized a “Train the Trainer’ session on Dec 26, during which the Farm Manager trained 5 local field officers on how to use the app’s “Capture” and “Diagnosis” features. This supported wider deployment before the final release in the following week.

2.5.2 Retrospect Release

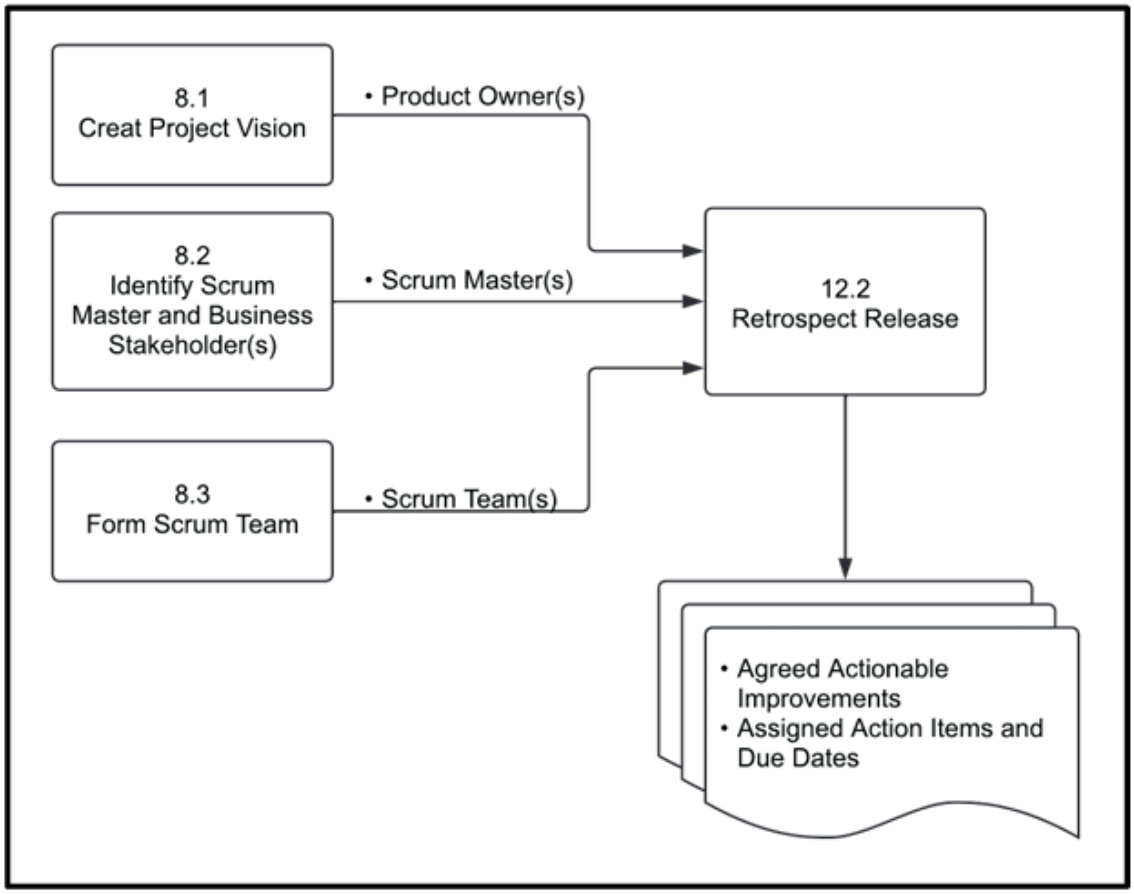


Figure 27. Data Flow Diagram for Retrospect Release Process (Satpathy, 2025)

2.5.2.1 Project Retrospective

The Retrospect Release process allowed the Scrum Team to pause and reflect on the entire project rather than just a single sprint. The goal was to identify broader strengths and weaknesses to improve the organization's future projects.

Key findings from the project's retrospective were as follows:

- **What Went Well:**
    - **Agility:** The decision to change to 50-point velocity weeks (Sprints 2 and 3) helped avoid project delays.
    - **Stakeholder Engagement:** The Farm Manager was a direct part of the Scrum Team. This allowed the User Feedback Loops to take only a few minutes instead of days.
- **What Needs Improvement:**
    - **Hardware Planning:** GPU memory crash confirmed that local hardware is not enough for modern AI training. Cloud GPUs need to be budgeted from day 1 in future projects.
    - **Connections Rates:** During planning, we assumed internet to be widespread across fields. Testing showed poor internet connections in some areas. Future updates must prioritize an "Offline Mode."

2.5.2.2 Release Agreement

The project was officially closed after signing of the Release Agreement between the Product Owner (Ali Mir) and the Project Sponsor (MRWO Director), confirming that all 8 User Stories were delivered, functioning as per the Acceptance Criteria.

CHAPTER 3: EXECUTION OF PROJECT ON JIRA

The 4 planned sprints have been completed. The details of the Jira execution are provided below.

3.1 Work Breakdown Structure

(See Annexure B)

3.2 Project Road Map and Agile Ceremony Schedule

The Jira view below shows the timeline of Epics. You can see the "Research" Epic completing in the first two weeks, followed by the "Development" Epic in the next 2 weeks.

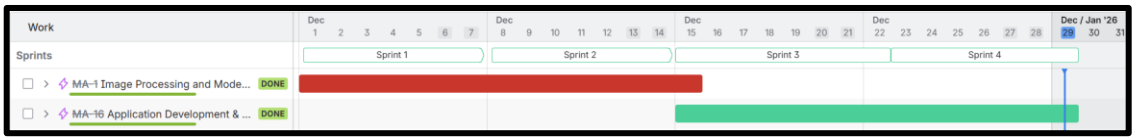


Figure 28. Jira Roadmap

The calendar below highlights the team's adherence to formal agile ceremonies. By conducting Sprint Planning, Daily Standup, Retrospective and Review meetings, the team optimized development time while ensuring consistent stakeholder feedback.

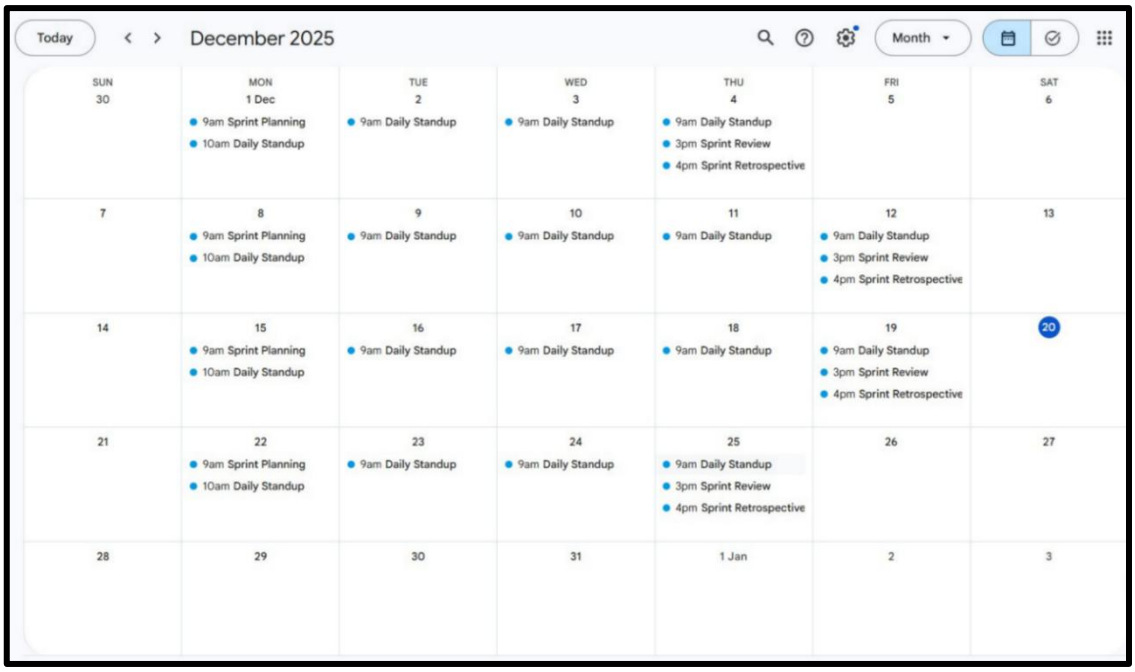


Figure 29. Agile Ceremony Schedule

Table 8. Agile Ceremony Details

Ceremony	**Frequency**	**Duration**	**Notes**
Sprint Planning	First working day of Sprint	2 Hours	Commit to Sprint Backlog.
Daily Standup	Every working day during Sprint	15 Mins	Synchronize on progress and blockers.
Grooming	Impromptu during sprint	Variable	Refine backlog as technical needs arise.
Sprint Review	Last working day of Sprint	1 Hour	Demonstrate working software to increment audience.
Sprint Retrospective	Last working day of Sprint	45 Mins	Continuous process improvement.

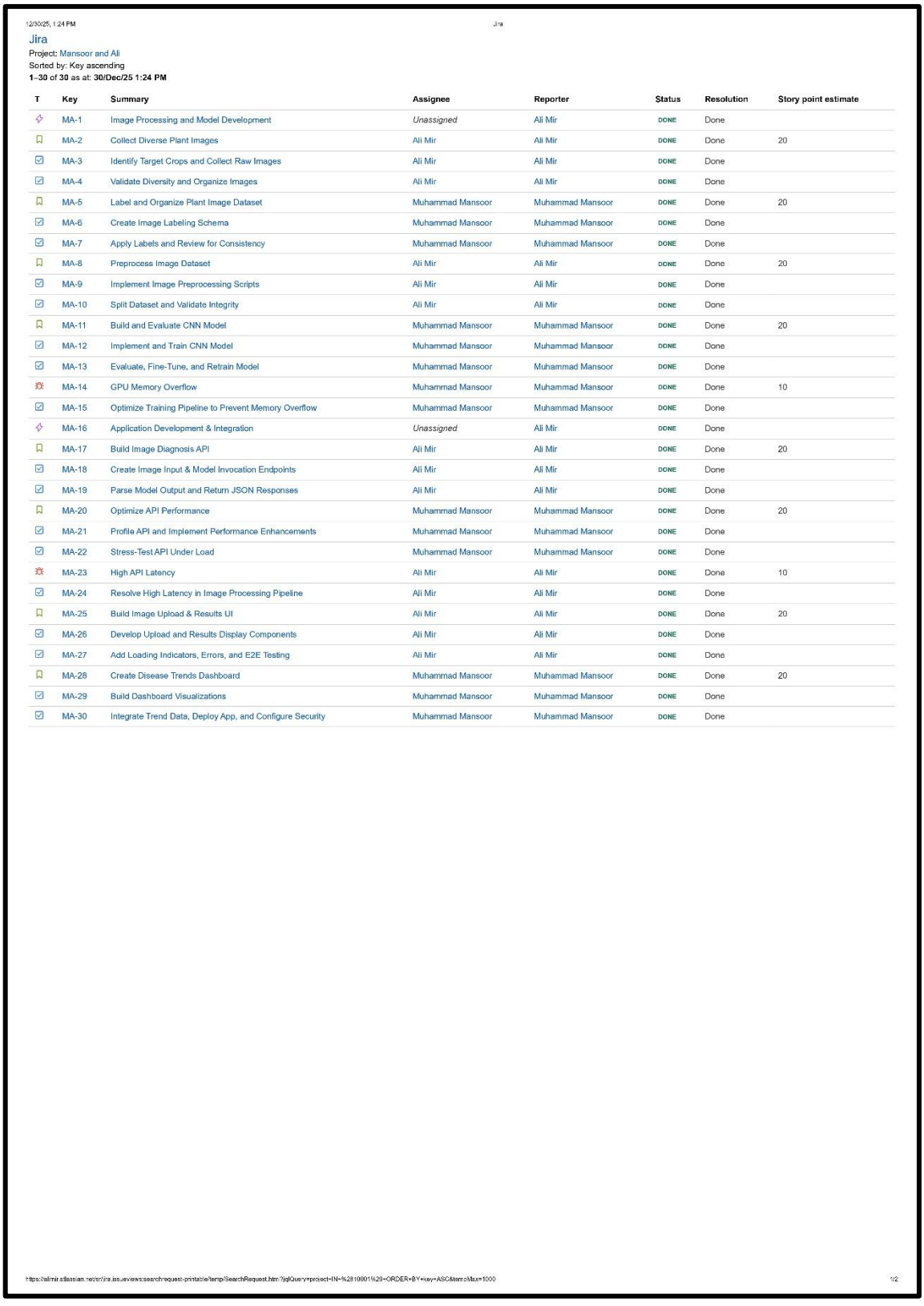
3.3 Product Backlog Status

The following image of the Product Backlog serves as a record of the project's Agile management. The backlog is structured into the following separate columns:

- The Key (e.g., MA-2, MA-5) provides a unique ID for traceability
- The Summary not only describes the requirement (e.g., "Collect Diverse Plant Images") but also identifies the work item type (Epic, User Story, Bug, or Task) to ensure clarity. The Story Points (e.g., 20) reflect the estimated effort needed to complete User Stories
- Status (e.g., Done) tracks current progress.

It is important to note the division of labor evidenced here:

- Execution of these tasks—such as the actual image collection, programming of the CNN model, and field testing—was performed by the subject matter experts (Agricultural Researcher, Software Engineer, Backend Developer, and Farm Manager), despite the assignees only being Mr. Ali Mir and Mr. Muhammad Mansoor.
- Entirety of the Jira project’s administration (and reporting) was done by Mr. Muhammad Mansoor and Mr. Ali Mir. They were exclusively responsible for organizing these boards, logging the user stories, tracking daily progress, and generating the status reports to ensure the project adhered to Scrum methodologies.



12/30/25, 1:24 PM Jira

Jira

Project: Mansoor and Ali
Sorted by: Key ascending
1–30 of **30** as at: **30/Dec/25 1:24 PM**

T	Key	Summary	Assignee	Reporter	Status	Resolution	Story point estimate
	MA-1	Image Processing and Model Development	*Unassigned*	Ali Mir	DONE	Done	
	MA-2	Collect Diverse Plant Images	Ali Mir	Ali Mir	DONE	Done	20
	MA-3	Identify Target Crops and Collect Raw Images	Ali Mir	Ali Mir	DONE	Done	
	MA-4	Validate Diversity and Organize Images	Ali Mir	Ali Mir	DONE	Done	
	MA-5	Label and Organize Plant Image Dataset	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	20
	MA-6	Create Image Labeling Schema	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	
	MA-7	Apply Labels and Review for Consistency	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	
	MA-8	Preprocess Image Dataset	Ali Mir	Ali Mir	DONE	Done	20
	MA-9	Implement Image Preprocessing Scripts	Ali Mir	Ali Mir	DONE	Done	
	MA-10	Split Dataset and Validate Integrity	Ali Mir	Ali Mir	DONE	Done	
	MA-11	Build and Evaluate CNN Model	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	20
	MA-12	Implement and Train CNN Model	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	
	MA-13	Evaluate, Fine-Tune, and Retrain Model	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	
	MA-14	GPU Memory Overflow	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	10
	MA-15	Optimize Training Pipeline to Prevent Memory Overflow	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	
	MA-16	Application Development & Integration	*Unassigned*	Ali Mir	DONE	Done	
	MA-17	Build Image Diagnosis API	Ali Mir	Ali Mir	DONE	Done	20
	MA-18	Create Image Input & Model Invocation Endpoints	Ali Mir	Ali Mir	DONE	Done	
	MA-19	Parse Model Output and Return JSON Responses	Ali Mir	Ali Mir	DONE	Done	
	MA-20	Optimize API Performance	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	20
	MA-21	Profile API and Implement Performance Enhancements	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	
	MA-22	Stress-Test API Under Load	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	
	MA-23	High API Latency	Ali Mir	Ali Mir	DONE	Done	10
	MA-24	Resolve High Latency in Image Processing Pipeline	Ali Mir	Ali Mir	DONE	Done	
	MA-25	Build Image Upload & Results UI	Ali Mir	Ali Mir	DONE	Done	20
	MA-26	Develop Upload and Results Display Components	Ali Mir	Ali Mir	DONE	Done	
	MA-27	Add Loading Indicators, Errors, and E2E Testing	Ali Mir	Ali Mir	DONE	Done	
	MA-28	Create Disease Trends Dashboard	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	20
	MA-29	Build Dashboard Visualizations	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	
	MA-30	Integrate Trend Data, Deploy App, and Configure Security	Muhammad Mansoor	Muhammad Mansoor	DONE	Done	

https://alimir.atlassian.net/sr/jira.issueviews:searchrequest-printable/temp/SearchRequest.html?jqlQuery=project+IN+%2810001%29+ORDER+BY+key+ASC&tempMax=1000 1/2

Figure 30. Product Backlog Status

3.4 Project Board Overview

The Active Sprint Board below shows how work items move from "To Do" to “In Progress” to "Done".

Figure 31. Jira Scrum Board

3.5 Project Bug Overview

The picture below shows the list of all Bugs (e.g., MA-14 GPU Overflow) and their resolution status.

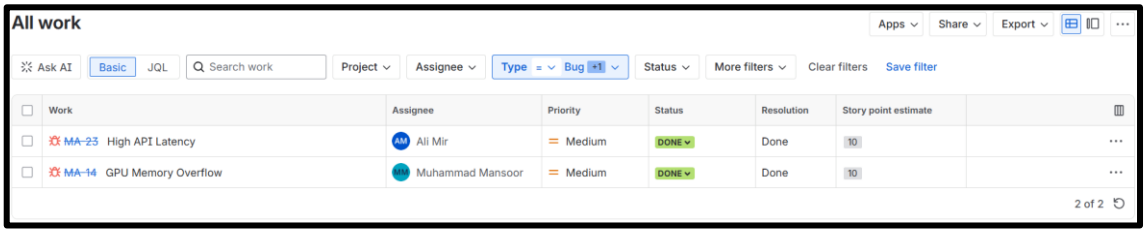


Figure 32. Project Bug Status

3.6 Sprint Burnup Chart

These charts show the total scope vs. completed work.

See report section 2.2.6.3: Sprint Burn Up and Burn Down Charts.

3.7 Sprint Burn Down Chart

These charts show the remaining effort decreasing as we approached the Dec 29th deadline.

See report section 2.2.6.3: Sprint Burn Up and Burn Down Charts.

3.8 Cumulative Flow Diagram

The Cumulative Flow Diagram (CFD) visualizes the steadiness of the project's workflow and the total distribution of work items across different statuses (to do, in progress, done) over time. The horizontal axis represents time (sprints), while the vertical axis represents the number of Story Points. The colored bands correspond to different workflow states:

- **Purple (To Do):** Represents the backlog of planned work.
- **Blue (In Progress):** Represents work currently being performed.
- **Green (Done):** Represents completed and accepted work.

For the MRWO Plant Doctor project, the CFD demonstrates a generally healthy flow with a consistent widening of the green "Done" band, indicating steady delivery of value. The parallel shape of the bands shows that a stable work-in-progress (WIP) limit was maintained throughout the sprints, signifying a predictable and efficient process leading up to the final release.

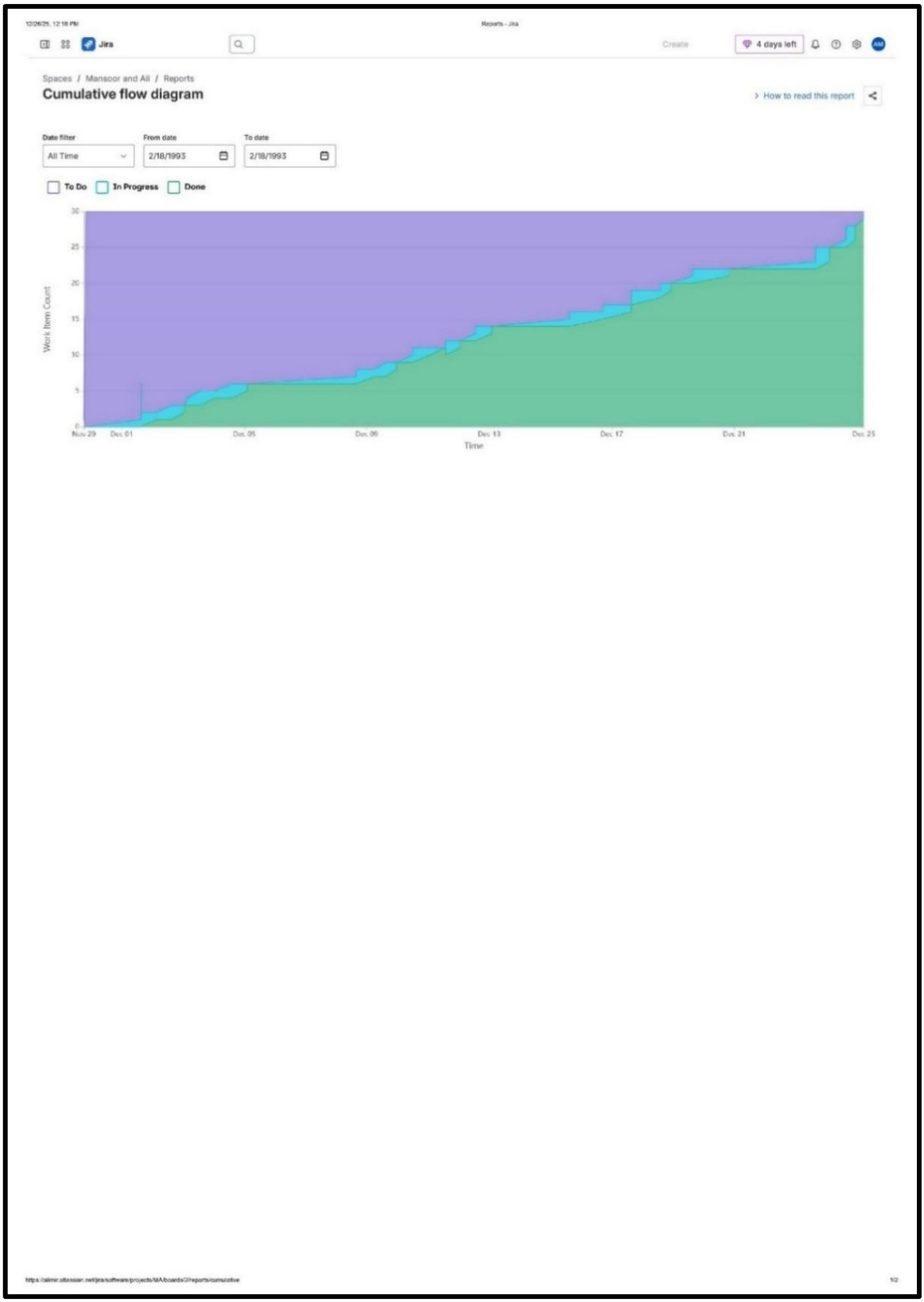


Figure 33. Cumulative Flow Diagram

3.9 Velocity Report

This chart demonstrates the adaptive capability of the team, where the baseline velocity of the team was 40 Story Points during regular sprints (Sprints 1 and 4) and 50 Story Points during weeks of intensive debugging (Sprints 2 and 3). The constant alignment between the gray 'Commitment' and the green 'Completed' bars verifies the fact that 100% of all planned work items have been delivered with no carryover.

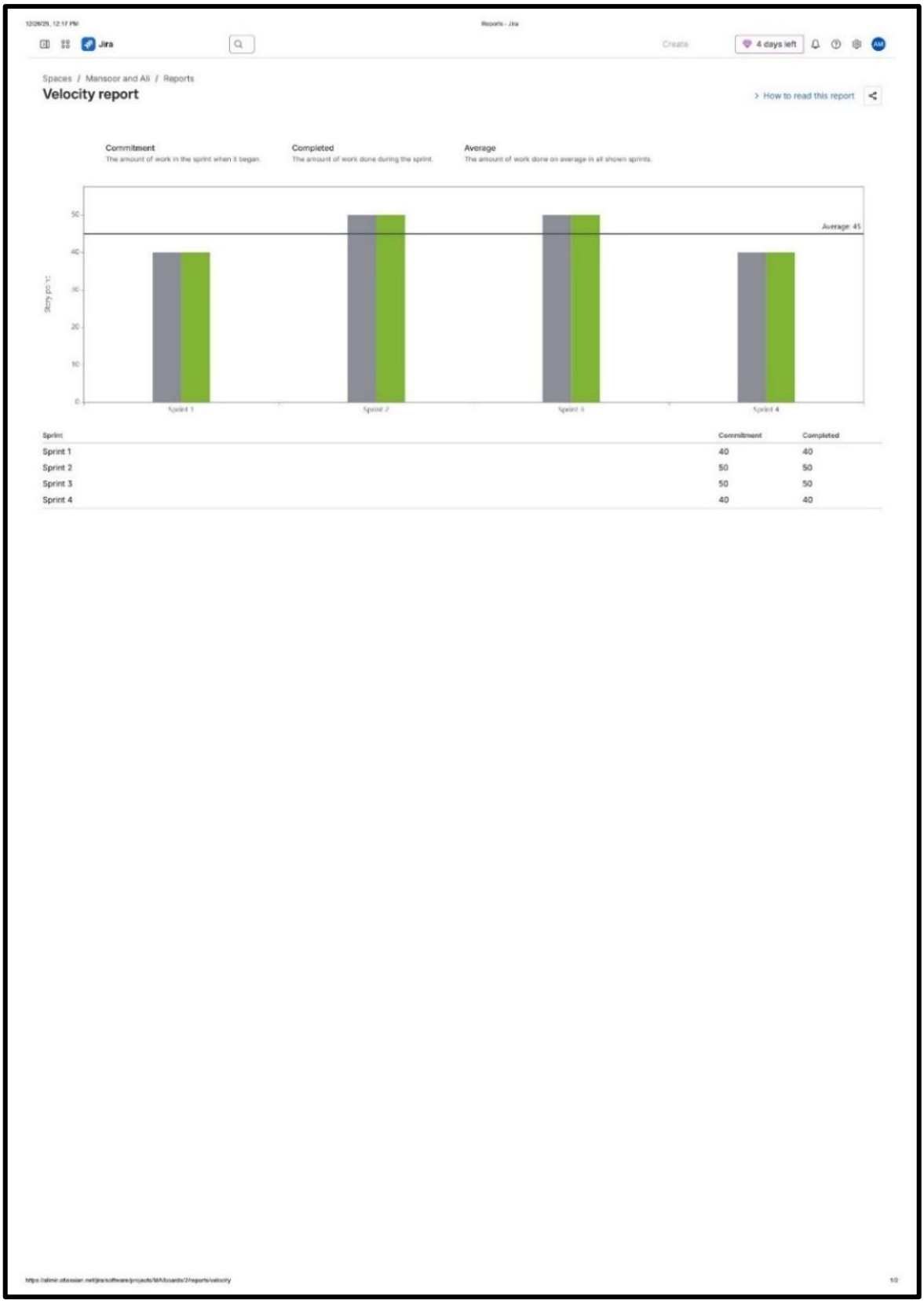


Figure 34. Velocity Report

REFERENCES

7 Tips for Writing Acceptance Criteria with Examples. (2017, May). Retrieved from agileforgrowth.com: https://agileforgrowth.com

Burn Up Vs Burn Down Charts. (2022, August). Retrieved from webopedia.com: https://www.webopedia.com

Don't overpay! How Much Does an App Development Cost. (2022, March). Retrieved from sumatosoft.com: https://sumatosoft.com/blog/how-much-does-mobile-app-development-cost

How does a Scrum Team Estimate and Commit to User Stories in a Scrum Project? (2014, June). Retrieved from scrumstudy.com: https://blog.scrumstudy.com

Satpathy, T. (2025). A Guide to the SCRUM BODY OF KNOWLEDGE (SBOK® Guide), Fifth Edition. SCRUMstudy.

The 3 Scrum Roles and Responsibilities Explained. (2023, June). Retrieved from coursera.org: https://www.coursera.org

The Sprint Planning Checklist. (2020, November). Retrieved from scrum.org: https://www.scrum.org

Top 5 User Story Estimation Techniques. (2022, March). Retrieved from 7pace.com: https://www.7pace.com

What are Epics and Features? (2023). Retrieved from scrum.org: https://www.scrum.org

What is a Definition of Done? (2023). Retrieved from scrum.org: https://www.scrum.org

What is Definition of Ready in Scrum? (2023). Retrieved from visual-paradigm.com: https://www.visual-paradigm.com

What is Product Owner: Scrum.org. (2023). Retrieved from Scrum.org: https://www.scrum.org

What is Scrum Master. (2023). Retrieved from Scrum.org: https://www.scrum.org

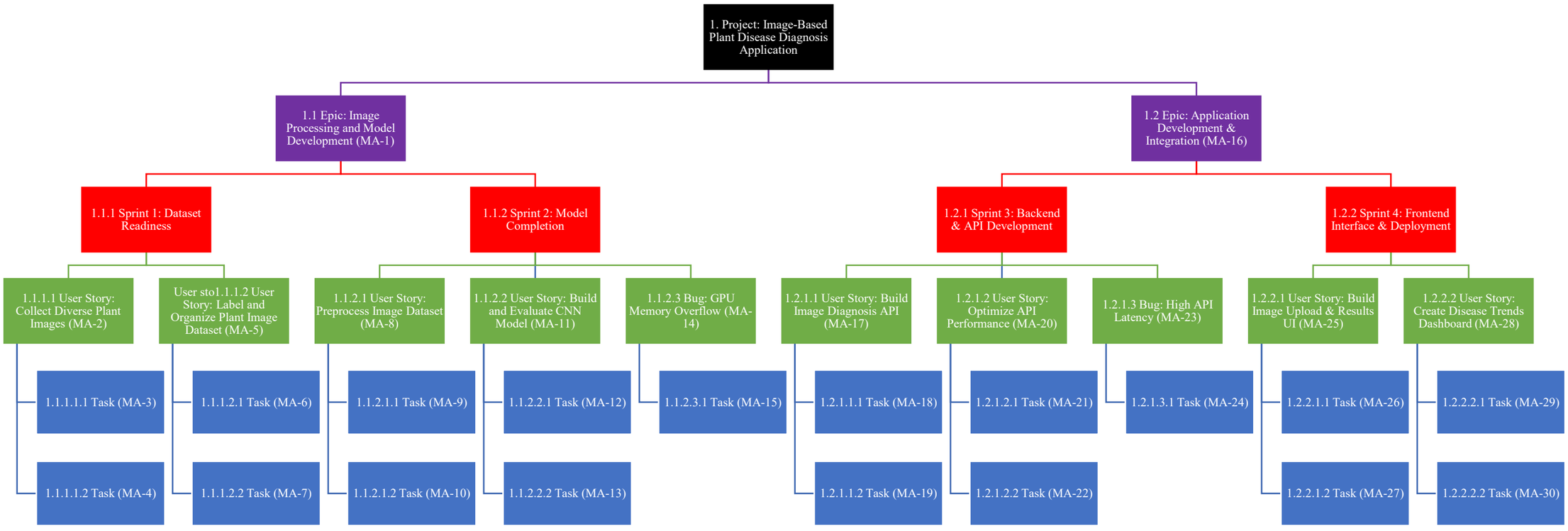
Who are key stakeholders. (2023). Retrieved from scrum.org: https://www.scrum.org

ANNEXURES

ANNEXURE A: COST BASELINE

Resource Role	**Rate (PKR/Day)**	**Days**	**Total Cost (PKR)**
Scrum Master	10,000	20	200,000
Product Owner	12,000	20	240,000
Ag Researcher	8,000	10	80,000
Software Engineer	15,000	20	300,000
Backend Dev	15,000	20	300,000
Farm Manager	5,000	5	25,000
Cloud/GPU Costs	Lump Sum	-	50,000
Field Logistics	Lump Sum	-	30,000
Contingency (10%)	-	-	122,500
GRAND TOTAL			**1,347,500**

ANNEXURE B: WORK BREAKDOWN STRUCTURE (WBS)



ANNEXURE C: RISK REGISTER

ID	Risk Description	Probability	Impact	Mitigation Strategy
R-1	**GPU Memory Overflow:** Model size surpasses hardware capacity.	High	High	Use batch processing; rent cloud GPU servers if local hardware fails.
R-2	**Unreliable Internet in Villages:** App fails to upload images.	Medium	High	Offline data synchronization and image compression.
R-3	**Incorrect Diagnosis:** AI mistakes healthy plant as diseased.	Low	High	Validate with high-quality Ground Truth data and add an advisory disclaimer.
R-4	**API Latency:** Server takes a long time to respond.	Medium	Medium	Keep model hot-loaded in memory; Use Caching.

END OF REPORT