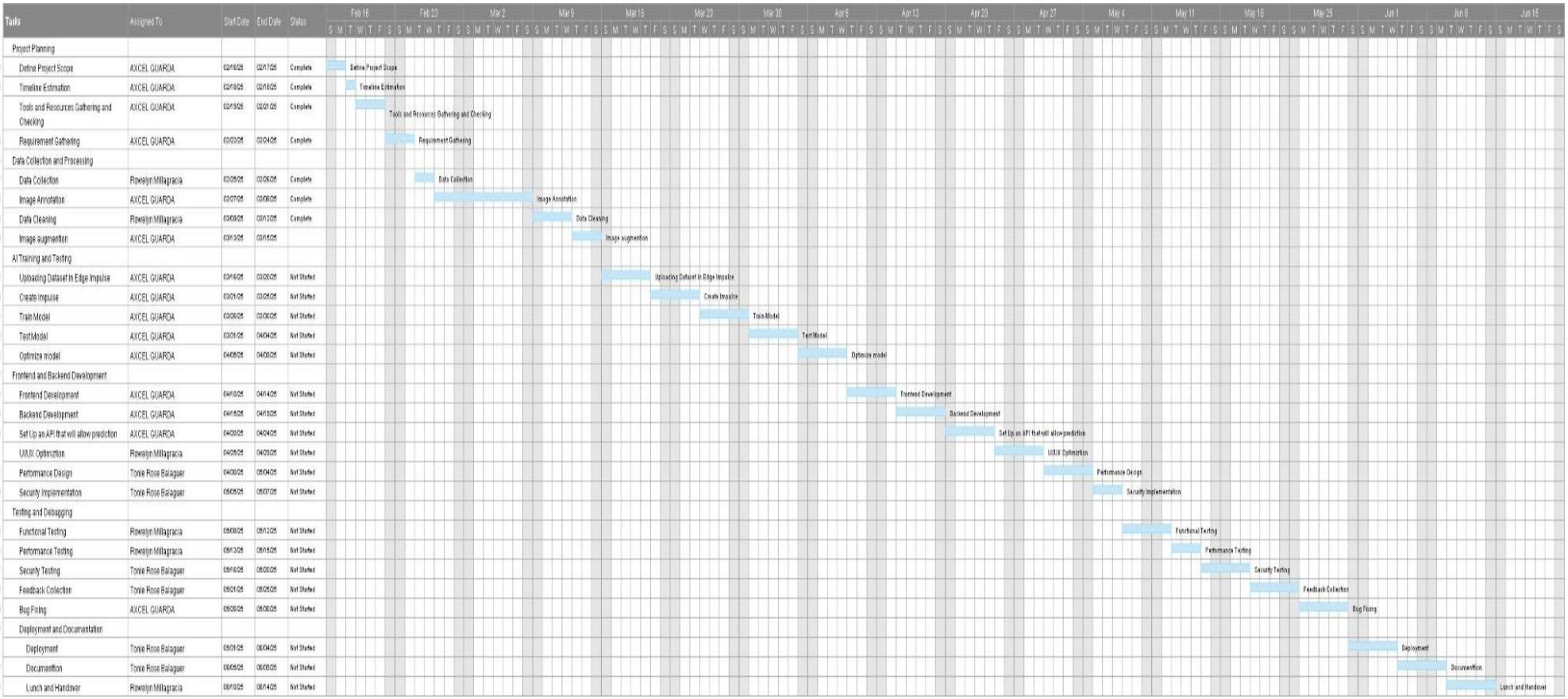
## Software Development Life Cycle (SDLC) Model

For the development of Palay Protector, the team adopted the Agile methodology as the Software Development Life Cycle (SDLC) model. Agile is a modern project management and software development approach that emphasizes flexibility, collaboration, and customer-centricity (Misra, Mittal, et al, 2024). The primary advantage of using Agile for Palay Protector lies in its adaptability. Unlike the Waterfall model, which follows a sequential process, Agile allows development to begin even if all requirements are not finalized. This makes it easier to introduce changes or fix errors as the project evolves, ensuring that the system meets user needs effectively. Agile also encourages internal feedback through team discussions rather than relying solely on external stakeholders, which is ideal for a project like Palay Protector that is designed to meet the needs of farmers.

Method: **Scrum**

Within the Agile framework, the team chose Scrum as the development method because it supports rapid iteration, teamwork, and transparency. Scrum's structure, centered around sprints, breaks down the project into smaller, manageable increments, allowing for continuous progress and frequent delivery of working features. Daily stand-up meetings help the team stay aligned, track progress effectively, and quickly address any blockers, which enhances productivity and project visibility. By adopting Scrum, the PalayProtector development team can deliver usable features early, incorporate feedback iteratively, and ensure the final product is reliable, user-focused, and meets the farmers' needs.

# GANTT CHART


Tasks
Assigned To
Start Date
End Date
Status
Feb 16
Feb 23
Mar 2
Mar 9
Mar 16
Mar 23
Mar 30
Apr 6
Apr 13
Apr 20
Apr 27
May 4
May 11
May 18
May 25
Jun 1
Jun 8
Jun 15
Project Planning
Define Project Scope AXCEL GUARDA 02/16/25 02/17/25 Complete
Timeline Estimation AXCEL GUARDA 02/16/25 02/16/25 Complete
Tools and Resources Gathering and Checking AXCEL GUARDA 02/19/25 02/21/25 Complete
Requirement Gathering AXCEL GUARDA 02/22/25 02/24/25 Complete
Data Collection and Processing
Data Collection Rowelyn Millagracia 02/26/25 02/26/25 Complete
Image Annotation AXCEL GUARDA 02/27/25 03/06/25 Complete
Data Cleaning Rowelyn Millagracia 03/06/25 03/12/25 Complete
Image augmention AXCEL GUARDA 03/13/25 03/16/25
AI Training and Testing
Uploading Dataset in Edge Impulse AXCEL GUARDA 03/16/25 03/20/25 Not Started
Create Impulse AXCEL GUARDA 03/21/25 03/26/25 Not Started
Train Model AXCEL GUARDA 03/26/25 03/30/25 Not Started
Test Model AXCEL GUARDA 03/31/25 04/04/25 Not Started
Optimize model AXCEL GUARDA 04/05/25 04/09/25 Not Started
Frontend and Backend Development
Frontend Development AXCEL GUARDA 04/10/25 04/14/25 Not Started
Backend Development AXCEL GUARDA 04/15/25 04/19/25 Not Started
Set Up an API that will allow prediction AXCEL GUARDA 04/20/25 04/24/25 Not Started
UI/UX Optimization Rowelyn Millagracia 04/25/25 04/29/25 Not Started
Performance Design Tonie Rose Balaguer 04/30/25 05/04/25 Not Started
Security Implementation Tonie Rose Balaguer 05/05/25 05/07/25 Not Started
Testing and Debugging
Functional Testing Rowelyn Millagracia 05/08/25 05/12/25 Not Started
Performance Testing Rowelyn Millagracia 05/13/25 05/16/25 Not Started
Security Testing Tonie Rose Balaguer 05/16/25 05/20/25 Not Started
Feedback Collection Tonie Rose Balaguer 05/21/25 05/25/25 Not Started
Bug Fixing AXCEL GUARDA 05/26/25 06/06/25 Not Started
Deployment and Documentation
Deployment Tonie Rose Balaguer 06/01/25 06/04/25 Not Started
Documention Tonie Rose Balaguer 06/05/25 06/09/25 Not Started
Lunch and Handover Rowelyn Millagracia 06/10/25 06/14/25 Not Started