

EO MAJI EO AFRICA EXPLORERS AGILE DEVELOPMENT PLAN

V1

Date: 25/04/2023

Contract No.

4000139395/22/I-DT

Submitted by



In Cooperation with:



UNIVERSITY OF
LEICESTER



DOCUMENT RELEASE SHEET

Author:	Radoslaw Guzinski (DHI), Michael Munk (DHI)	Sign: _____ Date: _____
Approval	Zoltan Szantoi	Sign: _____ Date: _____
Distribution:	ESA and partners	

CHANGE RECORD

Version	Date	Page(s)	Change record	Release
0	25/04/2023	10	First submission to ESA	1
1				
2				
3				

1 Introduction

1.1 Project objective

This project aims to implement a prototype for irrigation mapping and crop yield estimation using inputs from the scientific ECOSTRESS and PRISMA missions. The final aim is to develop workflows, in collaboration with the African Early Adopters and EO partner(s), that support African irrigation and food security management, as well as transferring these R&D learnings and results to African end-users and stakeholders. More specifically the project objectives in this project can overall be listed as:

- Exploration of the capabilities for future operational Copernicus missions (LSTM+CHIME) to estimate ET and crop water stress.
- Investigate the potential of PRISMA hyperspectral observations and thermal-based crop stress metrics to improve crop yield/biomass estimations to support agricultural monitoring
- Complement the ET retrievals with crop yield, in order to acquire a better understanding of water use efficiency (WUE) of cultivated landscapes.
- Direct involvement of African Early Adopters, in order to secure the usefulness and applicability of the prototype.
- Publish the findings in a freely available code repository and as scientifically peer-reviewed papers, as well as to promote the codes through other outreach activities such as development of digital notebooks.

All activities are to be carried out within the duration of the project lifetime from 1 December 2022 to 30 November 2024.

1.2 Scope of Document

This document presents the Agile Development Plan (PMP) which will be the formal, approved document used to guide agile prototype and toolbox development in the project “EO MAJI – EO Africa Explorers” (ESA AO/1-11038/21/I-DT).

1.3 Reference documents

REF-1	Statement of Work: ESA-EOP-SD-SOW-0250 – EO AFRICA EXPLORERS
REF-2	EO MAJI proposal dated 18/02/2022
REF-3	Clarification request from ESA dated 06/06/2022
REF-4	Response to clarification dated 22/06/2022
REF-5	Contract No. 4000139395/22/I-DT

2 Principles of Agile Development

2.1 Introduction to Agile Development

Agile Development is an approach to software development that emphasizes flexibility, collaboration, and end-user satisfaction. Agile is a popular approach used by software development teams to deliver high-quality products quickly and efficiently through a set of incremental, iterative practices. The Agile Development Plan is a detailed blueprint that guides the development team through the software development process and outlines the project's objectives, timelines, and resources needed to achieve the project goals. This document provides insights into the Agile methodology and concrete tools for implementation by describing the fundamental principles of and benefits of Agile Development, key components of an Agile Development Plan, and the practice of Continuous Integration and Continuous Testing (CI/CT) that is incorporated to the EO-MAJI project.

Agile Development is based on a set of fundamental principles that guide its philosophy and practices. The four core values that guide Agile Development are as follows:

1. **Individuals and interactions over processes and tools** – Agile Development emphasizes the importance of collaboration and communication between team members over relying solely on processes and tools. It values the insights and contributions of all team members and promotes a culture of open communication.
2. **Working software over comprehensive documentation** – Agile Development prioritizes delivering a working product over creating extensive documentation. It values frequent releases and feedback, allowing for continuous improvement and adaptation.
3. **End-user collaboration over contract negotiation** – Agile Development involves continuous collaboration with end-users throughout the development process, ensuring that the product meets their needs and expectations. It values end-user satisfaction over strict adherence to contracts.
4. **Responding to change over following a plan** – Agile Development is adaptable and responsive to change, with the ability to adjust plans and priorities as needed. It values flexibility and the ability to pivot quickly in response to new information or changing requirements.

By following these principles, the EO-MAJI team promotes a collaborative, adaptable, and end-user-focused approach to software development, resulting in a more efficient and effective development process. This ensures a iterative software development cycle (see)

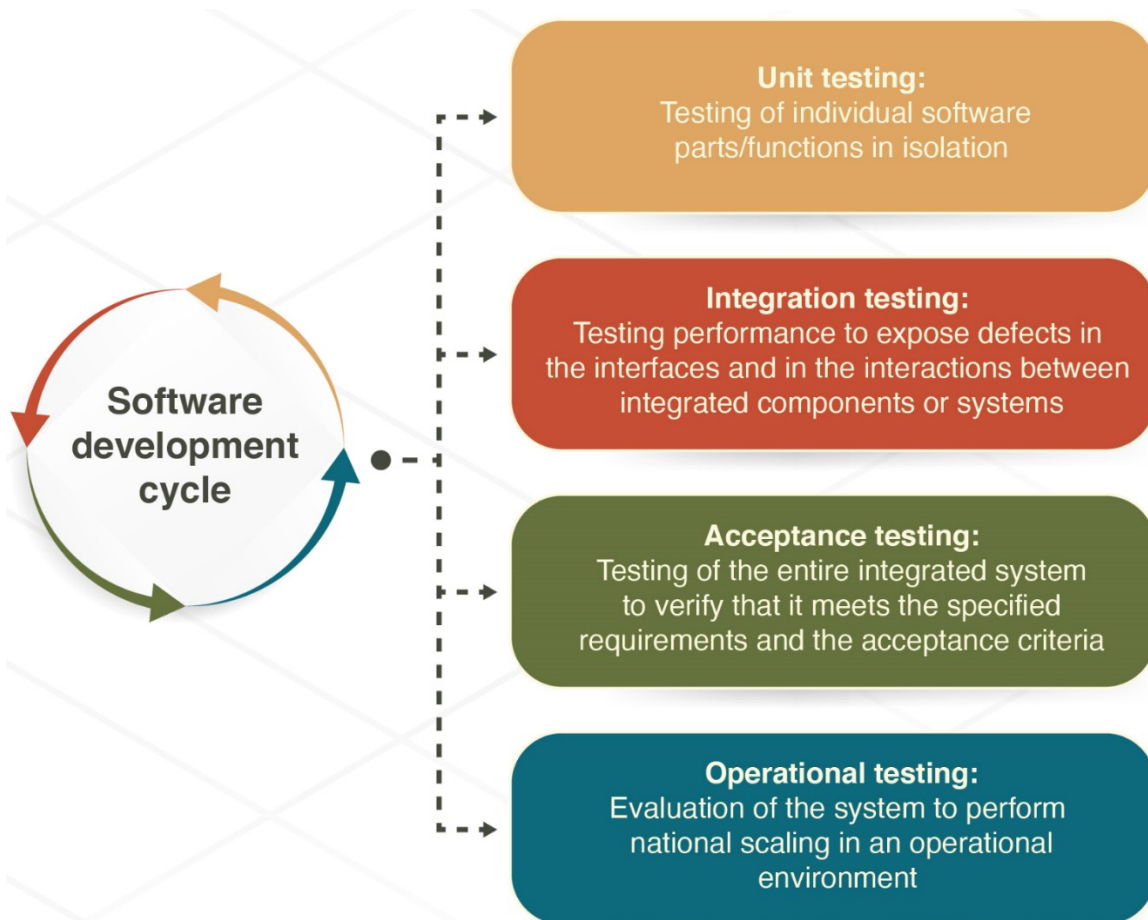


Figure 1 Implemented software development cycle testing with progressive check points to ensure high quality and transferability.

2.2. Benefits of Agile Development

Agile Development offers many benefits to software and algorithm development teams and their organizations that can improve the efficiency, quality, and end-user satisfaction of software development projects. Some of the key benefits include:

1. **Faster time-to-market:** Agile Development emphasizes the delivery of working software in small increments, enabling faster feedback and iteration. This allows teams to release products faster, giving organizations a competitive advantage in the market.
2. **Improved quality:** Agile Development practices such as continuous integration and automated testing promote the production of high-quality software. This leads to fewer defects and a more reliable product.
3. **Increased collaboration and communication:** Agile Development emphasizes collaboration and communication among team members, stakeholders, and end-users. This promotes a better understanding of requirements and a more efficient development process.
4. **Enhanced flexibility and adaptability:** Agile Development enables teams to be more responsive to changes in requirements or market conditions, enabling them to adjust their plans and priorities quickly.

5. **Improved end-user satisfaction:** Agile Development prioritizes continuous collaboration with end-users throughout the development process. This ensures that the product meets the end-user's needs and expectations, resulting in a higher level of end-user satisfaction and loyalty.
6. **Reduced risk:** Agile Development methodology allows teams to identify and mitigate potential setbacks earlier in the development process through its focus on iterative, incremental progress. This results in a reduced risk of project failure or delays.

2.3 Agile Development Plan

By adapting an Agile Development Plan EO-MAJI ensures is a flexible and adaptive plan that guides the development team through the software development process. It consists of several key components that facilitate collaboration and communication among team members and stakeholders. The essential components of an Agile Development Plan are:

1. **Product vision:** The product vision defines the long-term goals and objectives of the software product. It sets the direction and purpose of the development effort, providing a clear understanding of the product's intended outcome.
2. **User stories:** User stories are short, simple descriptions of a feature or functionality from the perspective of the user. They describe what the user needs or wants and why, providing a clear understanding of the user's requirements.
3. **Backlog:** The backlog is a prioritized list of user stories and tasks that need to be completed to deliver the product. It includes both functional and non-functional requirements and is continuously updated throughout the development process.
4. **Sprint planning:** Sprint planning is a collaborative process that involves the development team and stakeholders in planning the work to be done during a sprint—the period during which development occurs. This planning process involves defining the sprint goal, selecting user stories from the backlog, and estimating the effort required to complete them.
5. **Sprint backlog:** The sprint backlog is a subset of the overall backlog and includes the user stories and tasks that the development team has committed to completing during the current sprint.
6. **Daily stand-up:** The daily stand-up is a short, daily meeting where the development team updates each other on progress, discusses any obstacles, and plans the work for the day.
7. **Retrospective:** The retrospective is a meeting held at the end of each sprint to reflect on the development process and identify areas for improvement.

Effective implementation of these key components ensures that EO-MAJI team members are working towards a shared vision and have a clear understanding of the work to be done. This promotes collaboration, communication, and adaptability, resulting in a more streamlined and productive development process. The following sections describe three of the crucial components of an Agile Development Plan—user stories, sprint planning, and retrospectives—in greater detail.

2.3.1 User stories

User stories play a crucial role in Agile Development as they serve as the primary means of capturing and communicating user requirements. User stories are short, simple descriptions of a feature or functionality from the perspective of the software user. They describe what the user needs (the early

adopters) or wants and why, providing a clear understanding of the user's requirements which translate into tasks for the development team. User stories typically follow a simple format, such as:

"As a [user], I want to [goal/objective], so that [benefit]."

For example, "As an end-user, I want to be able to search for products by category, so that I can find what I'm looking for quickly."

User stories are used to guide the development team's work throughout the development process. They are the basis for the product backlog, which is a prioritized list of user stories that need to be completed to deliver the product. The development team works on user stories in order of priority, completing each one before moving on to the next.

During sprint planning, the development team and stakeholders review and prioritize the user stories in the backlog, selecting the ones that will be completed during the upcoming sprint. This ensures that the development effort is focused on delivering the most valuable features first.

User stories help to ensure that the development team is focused on meeting the needs of the user, which is central to Agile Development's end-user-centric approach to software development. This leads to a more effective and efficient development process, resulting in a higher quality product and increased end-user satisfaction.

The EO-MAJI project team will collaborate closely with the Early adopters and local EO collaborators and iteratively update and concretise the user stories in order to steer the agile progress of the project.

2.3.2 Sprint planning

Sprint planning is a component of Agile Development that plays a crucial role in guiding the development team's work. It is a collaborative process that involves the development team and stakeholders in planning the work to be done during a sprint, which is the period— typically lasting two weeks— during which potentially shippable products are developed. Sprint planning takes place on the first day of a new sprint, and its primary role is to ensure that the development team has a clear understanding of what needs to be done and how it will be accomplished.

Some of the key roles of sprint planning in Agile Development include:

1. **Defining the sprint goal:** During sprint planning, the development team and stakeholders work together to define the sprint goal, which is a short statement that describes the objective of the sprint and what the development team hopes to achieve. The sprint goal helps to guide the development team's work and provides a clear focus for the sprint.
2. **Selecting user stories:** The development team and stakeholders review the product backlog and select the user stories that will be completed during the sprint. The selection process is based on the priority of the user stories and their relevance to the sprint goal.
3. **Estimating effort:** The development team estimates the effort required to complete each user story. This helps to ensure that the team is committing to a realistic amount of work for the sprint.
4. **Planning the work:** The development team plans the work required to complete the selected user stories, breaking them down into smaller tasks as necessary. The team identifies

dependencies between tasks and considers the capacity of the team to complete the work within the sprint.

5. **Creating the sprint backlog:** The sprint backlog is a subset of the product backlog and includes the user stories and tasks that the development team has committed to completing during the sprint. The sprint backlog serves as a plan for the development team's work during the sprint.

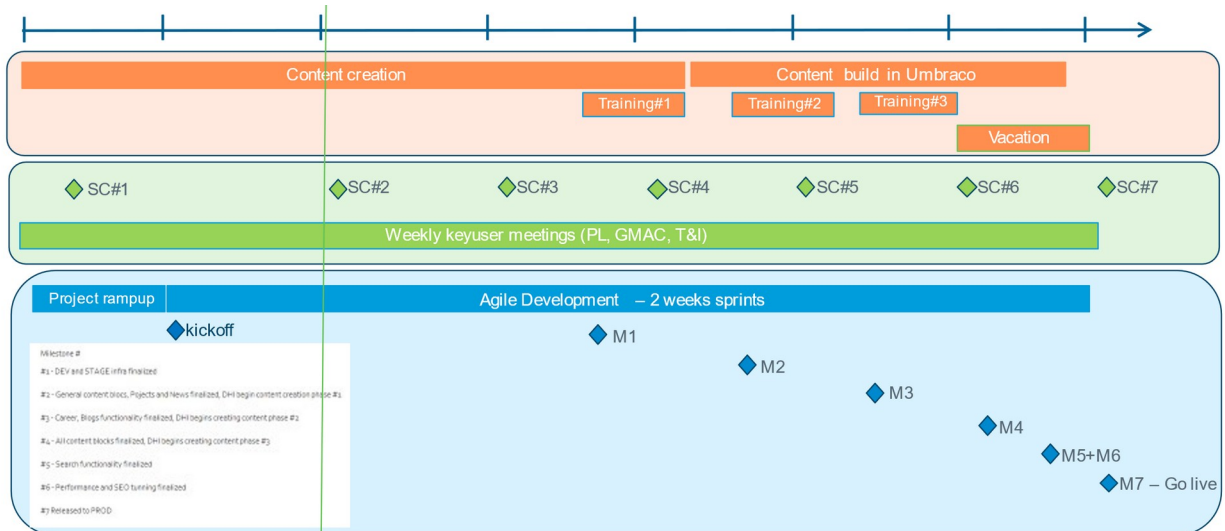


Figure 2 example of elements layout implemented in software and algorithm development.

By engaging in sprint planning, the EO-MAJI development team and stakeholders will ensure that everyone has a clear understanding of the sprint goals and what needs to be done to achieve them. This promotes collaboration and communication, helping to ensure that the development team is working efficiently and effectively towards delivering high-quality software.

2.3.3 Retrospectives

Retrospective meetings play a vital role in Agile Development as they provide the development team with an opportunity to reflect on their work, identify areas for improvement, and make changes to the development process. Retrospectives are held at the end of each sprint and involve the entire development team, with a facilitator leading the discussion.

The main role of retrospectives in Agile Development includes:

1. **Reflection:** Retrospectives allow the development team to reflect on the work that was completed during the sprint. This reflection helps the team to identify what they did well, any issues or challenges that arose, and where they can improve in the future.
2. **Continuous improvement:** Retrospectives are an essential tool for continuous improvement in Agile Development. The development team identifies areas for improvement, such as process inefficiencies or communication breakdowns, and works together to address them.
3. **Feedback loop:** Retrospectives provide a feedback loop for the development team and stakeholders. By reflecting on the sprint, the development team can provide feedback to stakeholders on the progress of the project and identify any changes that need to be made to the product backlog.

4. **Team building:** Retrospectives provide an opportunity for the development team to come together and discuss their work. This promotes collaboration, communication, and teamwork, leading to a more effective and efficient development process.
5. **Agile mindset:** Retrospectives help to reinforce the Agile mindset of continuous improvement and adaptation. By regularly reflecting on their work, the development team can learn from their mistakes and make changes to the development process to improve the quality of the software they deliver.

Incorporating retrospectives into the EO-MAJI development process allows teams to ensure that they are delivering high-quality software and continuously improving their development process.

2.4 Continuous integration and testing

Agile Development emphasizes the delivery of high-quality products through iterative, incremental progress. The practice of Continuous integration and testing (CI/CT) plays a critical role in achieving this by ensuring that software is developed and delivered in a consistent and high-quality manner. The primary role of CI/CT is to continuously integrate frequent code changes made by the development team and ensure that the code is tested automatically. Implementing these practices enables development teams to efficiently deliver software that meets the needs of their stakeholders and users.

The key roles of continuous integration and testing in Agile Development include:

1. **Early detection of issues:** CI/CT helps to detect issues early in the development process, allowing the development team to address them quickly. By catching problems early, the team can reduce the amount of time and effort required to fix them.
2. **Faster feedback loop:** CI/CT provides a faster feedback loop to the development team, allowing them to receive immediate feedback on the code changes they have made. This feedback loop helps to identify issues early, allowing the team to address them before they become more significant problems.
3. **Improved quality:** CI/CT helps to improve the overall quality of the software by ensuring that code changes are tested thoroughly and integrated correctly. This leads to a more stable and reliable codebase.
4. **Efficiency:** CI/CT helps to increase the efficiency of the development process by automating many of the testing and integration tasks that would otherwise be done manually. This allows the development team to focus on other critical tasks, such as developing new features and fixing bugs.
5. **Collaboration:** CI/CT promotes collaboration among the development team by providing a common platform for testing and integration. This helps to ensure that everyone is working from the same codebase and that any issues are identified and resolved quickly.

During Continuous Integration (CI), several types of testing are typically performed to ensure that the code changes made by the development team are integrated correctly and function as expected. These tests are automated and run continuously as part of the CI process, providing immediate feedback to the development team.

Some of the key types of testing that are performed during Continuous Integration include:

1. **Unit testing:** Unit testing is a type of testing that focuses on individual units of code, such as functions or methods, and is quick to implement. It verifies that each unit of code is working as intended and identifies any errors or issues.
2. **Integration testing:** Integration testing verifies that different units of code can work together correctly when integrated. It tests the interactions between different components of the system and ensures that they are working as expected.
3. **Acceptance testing:** Acceptance testing verifies that the software meets the functional requirements specified by the stakeholders. It ensures that the software functions as intended and that it meets the needs of the users.
4. **Performance testing:** Performance testing verifies that the software meets the performance requirements specified by the stakeholders in terms of speed, reliability, or scalability for example. It ensures that the software can handle the expected load and that it performs optimally.
5. **Security testing:** Security testing verifies that the software is secure and that it meets the security requirements specified by the stakeholders. It ensures that the software is protected against potential security threats.

The goal of Continuous Integration testing is to ensure that EO-MAJI product code changes are tested thoroughly and integrated correctly, and that the software functions as intended. By performing these tests continuously, the development team can identify issues early and ensure that the software is of high quality.